

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

**INFORME DE LABORATORIO No 01**

**CURSO:**

BASE DE DATOS II

**DOCENTE:**

Ing. Patrick Cuadros Quiroga

Integrantes:

|                                       |              |
|---------------------------------------|--------------|
| Espinoza Caso, Lisbeth                | (2011040667) |
| Salamanca Contreras, Fiorella Rosmery | (2015053237) |
| Condori Gutierrez, Flor de Maria      | (2015053227) |

# Índice

|   |    |
|---|----|
| 1. Actividad No 01 – Revisión de Sintaxis       | 1  |
| 2. Actividad No 02 – Reconociendo la estructura | 3  |
| 3. Actividad No 03 – Consultas Básicas          | 6  |
| 4. Actividad No 04 – Restricción y Ordenamiento | 9  |
| 5. Actividad No 05 – Funciones                  | 17 |
| 6. Actividad No 06 – Funciones de Conversión    | 24 |
| 7. Actividad No 07 – Funciones de Agrupación    | 28 |
| 8. Actividad No 08 – Enlaces                    | 30 |
| 9. Actividad No 09 – SubConsultas               | 38 |
| 10. Actividad No 10 – Conjuntos                 | 46 |

# 1. Actividad No 01 – Revisión de Sintaxis

De los siguientes comandos ¿Cuál es el resultado? ¿En caso de ser error cual sería la sentencia correcta?

- SELECT last\_name, job\_id, salary AS Sal FROM employees;  
Es correcta

```
SELECT last_name, job_id, salary AS Sal FROM employees;  
go
```

|    | last_name | job_id    | Sal      |
|----|-----------|-----------|----------|
| 1  | King      | AD_PRES   | 24000.00 |
| 2  | Kochhar   | AD_VP     | 17000.00 |
| 3  | De Haan   | AD_VP     | 17000.00 |
| 4  | Hunold    | IT_PROG   | 9000.00  |
| 5  | Ernst     | IT_PROG   | 6000.00  |
| 6  | Austin    | IT_PROG   | 4800.00  |
| 7  | Pataballa | IT_PROG   | 4800.00  |
| 8  | Lorentz   | IT_PROG   | 4200.00  |
| 9  | Greenberg | FI_MGR    | 12008.00 |
| 10 | Faviet    | FI_ACC... | 9000.00  |
| 11 | Chen      | FI_ACC... | 8200.00  |
| 12 | Sciarra   | FI_ACC... | 7700.00  |
| 13 | Uman      | FI_ACC... | 7800.00  |
| 14 | Popp      | FI_ACC... | 6900.00  |
| 15 | Raphaely  | PU_MAN    | 11000.00 |

- SELECT \* FROM job\_grades;  
Es incorrecta, la sentencia correcta sería:

SELECT \* FROM jobs;

```
SELECT * FROM jobs;  
go
```

|    | job_id     | job_title                       | min_salary | max_salary |
|----|------------|---------------------------------|------------|------------|
| 1  | AC_ACCOUNT | Public Accountant               | 4200       | 9000       |
| 2  | AC_MGR     | Accounting Manager              | 8200       | 16000      |
| 3  | AD_ASST    | Administration Assistant        | 3000       | 6000       |
| 4  | AD PRES    | President                       | 20080      | 40000      |
| 5  | AD_VP      | Administration Vice President   | 15000      | 30000      |
| 6  | FI_ACCOUNT | Accountant                      | 4200       | 9000       |
| 7  | FI_MGR     | Finance Manager                 | 8200       | 16000      |
| 8  | HR_REP     | Human Resources Representative  | 4000       | 9000       |
| 9  | IT_PROG    | Programmer                      | 4000       | 10000      |
| 10 | MK_MAN     | Marketing Manager               | 9000       | 15000      |
| 11 | MK_REP     | Marketing Representative        | 4000       | 9000       |
| 12 | PR_REP     | Public Relations Representative | 4500       | 10500      |
| 13 | PU_CLERK   | Purchasing Clerk                | 2500       | 5500       |
| 14 | PU_MAN     | Purchasing Manager              | 8000       | 15000      |
| 15 | SA_MAN     | Sales Manager                   | 10000      | 20080      |
| 16 | SA_REP     | Sales Representative            | 6000       | 12008      |
| 17 | SH_CLERK   | Shipping Clerk                  | 2500       | 5500       |
| 18 | ST_CLERK   | Stock Clerk                     | 2008       | 5000       |
| 19 | ST_MAN     | Stock Manager                   | 5500       | 8500       |

- SELECT employee\_id, last\_name sal x 12 ANNUAL SALARY FROM employees;  
Es incorrecta, la sentencia correcta sería:

SELECT employee\_id, last\_name, salary \* 12 'ANNUAL SALARY' FROM employees;

```
~
SELECT employee_id, last_name, salary*12 'ANNUAL SALARY' FROM employees;
go|
```

|    | employee_id | last_name | ANNUAL SALARY |
|----|-------------|-----------|---------------|
| 1  | 100         | King      | 288000.00     |
| 2  | 101         | Kochhar   | 204000.00     |
| 3  | 102         | De Haan   | 204000.00     |
| 4  | 103         | Hunold    | 108000.00     |
| 5  | 104         | Ernst     | 72000.00      |
| 6  | 105         | Austin    | 57600.00      |
| 7  | 106         | Pataballa | 57600.00      |
| 8  | 107         | Lorentz   | 50400.00      |
| 9  | 108         | Greenberg | 144096.00     |
| 10 | 109         | Faviet    | 108000.00     |
| 11 | 110         | Chen      | 98400.00      |
| 12 | 111         | Sciarra   | 92400.00      |
| 13 | 112         | Uman      | 93600.00      |
| 14 | 113         | Popp      | 82800.00      |
| 15 | 114         | Raphaely  | 132000.00     |

## 2. Actividad No 02 – Reconociendo la estructura

1. Se requiere determinar la estructura de la tabla DEPARTMENTS y sus datos.

SP\_HELP 'DEPARTMENTS'

SP\_HELP departments;

go

Results

Messages

|   | Name        | Owner | Type       | Created_datetime        |
|---|-------------|-------|------------|-------------------------|
| 1 | departments | dbo   | user table | 2018-09-28 21:48:06.947 |

|   | Column_name     | Type    | Computed | Length | Prec | Scale | Nullable | TrimTrailingBlanks | FixedLenNullInSource | Collation                    |
|---|-----------------|---------|----------|--------|------|-------|----------|--------------------|----------------------|------------------------------|
| 1 | department_id   | int     | no       | 4      | 10   | 0     | no       | (n/a)              | (n/a)                | NULL                         |
| 2 | department_name | varchar | no       | 30     |      |       | yes      | no                 | yes                  | SQL_Latin1_General_CP1_CI_AS |
| 3 | manager_id      | int     | no       | 4      | 10   | 0     | yes      | (n/a)              | (n/a)                | NULL                         |
| 4 | location_id     | int     | no       | 4      | 10   | 0     | yes      | (n/a)              | (n/a)                | NULL                         |

|   | Identity                    | Seed | Increment | Not For Replication |
|---|-----------------------------|------|-----------|---------------------|
| 1 | No identity column defined. | NULL | NULL      | NULL                |

|   | RowGuidCol                    |
|---|-------------------------------|
| 1 | No rowguidcol column defined. |

|   | Data_located_on_filegroup |
|---|---------------------------|
| 1 | PRIMARY                   |

|   | index_name | index_description                                 | index_keys    |
|---|------------|---|---------------|
| 1 | dept_id_pk | clustered, unique, primary key located on PRIMARY | department_id |

|   | constraint_type         | constraint_name | delete_action | update_action | status_enabled | status_for_replication | constraint_keys                                     |
|---|-------------------------|-----------------|---------------|---------------|----------------|------------------------|---|
| 1 | PRIMARY KEY (clustered) | dept_id_pk      | (n/a)         | (n/a)         | (n/a)          | (n/a)                  | department_id                                       |
| 2 | FOREIGN KEY             | dept_loc_fk     | No Action     | No Action     | Enabled        | Is_For_Replication     | location_id   |
| 3 |                         |                 |               |               |                |                        | REFERENCES Human_Resources.dbo.locations (locati... |

|   | Table is referenced by foreign key             |
|---|--|
| 1 | Human_Resources.dbo.employees: emp_dept_fk     |
| 2 | Human_Resources.dbo.job_history: jhist_dept_fk |

2. El departamento de Recursos Humanos requiere un reporte que muestre los campos: employee\_id, last\_name y job\_id, asicomo el campo hire\_date con el alias StartDate.

```
SELECT emp.employee_id,
emp.last_name,
emp.job_id,
emp.hire_date AS StartDate
FROM employees AS emp;
```

```
SELECT emp.employee_id,
emp.last_name,
emp.job_id,
emp.hire_date AS StartDate
FROM employees AS emp;
go
```

| Results |             | Messages  |            |            |
|---------|-------------|-----------|------------|------------|
|         | employee_id | last_name | job_id     | StartDate  |
| 1       | 100         | King      | AD_PRES    | 2003-06-17 |
| 2       | 101         | Kochhar   | AD_VP      | 2005-09-21 |
| 3       | 102         | De Haan   | AD_VP      | 2001-01-13 |
| 4       | 103         | Hunold    | IT_PROG    | 2006-01-03 |
| 5       | 104         | Ernst     | IT_PROG    | 2007-05-21 |
| 6       | 105         | Austin    | IT_PROG    | 2005-06-25 |
| 7       | 106         | Pataballa | IT_PROG    | 2006-02-05 |
| 8       | 107         | Lorentz   | IT_PROG    | 2007-02-07 |
| 9       | 108         | Greenberg | FI_MGR     | 2002-08-17 |
| 10      | 109         | Faviet    | FI_ACCOUNT | 2002-08-16 |
| 11      | 110         | Chen      | FI_ACCOUNT | 2005-09-28 |
| 12      | 111         | Sciarra   | FI_ACCOUNT | 2005-09-30 |
| 13      | 112         | Uman      | FI_ACCOUNT | 2006-03-07 |
| 14      | 113         | Popp      | FI_ACCOUNT | 2007-12-07 |
| 15      | 114         | Raphaely  | PU_MAN     | 2002-12-07 |

3. Finalmente el departamento de Recursos Humanos requiere un listado de todos valores del campo JOB\_ID de la tabla EMPLOYEES pero que se muestren de forma única y no repetida.

```
SELECT DISTINCT job_id FROM employees;
```

```
SELECT DISTINCT job_id FROM employees;
go
```

| Results |            | Messages |
|---------|------------|----------|
|         | job_id     |          |
| 1       | AC_ACCOUNT |          |
| 2       | AC_MGR     |          |
| 3       | AD_ASST    |          |
| 4       | AD PRES    |          |
| 5       | AD_VP      |          |
| 6       | FI_ACCOUNT |          |
| 7       | FI_MGR     |          |
| 8       | HR_REP     |          |
| 9       | IT_PROG    |          |
| 10      | MK_MAN     |          |
| 11      | MK_REP     |          |
| 12      | PR_REP     |          |
| 13      | PU_CLERK   |          |
| 14      | PU_MAN     |          |
| 15      | SA_MAN     |          |

### 3. Actividad No 03 – Consultas Básicas

1. El departamento de Recursos Humanos requiere ampliar el reporte anterior (4.2.2) para hacerlo más comprensible, por lo que se requiere que los encabezados de las columnas sean: Emp No, Empleado, Puesto y Fecha Contratación.

```
SELECT emp.employee_id AS 'Emp N',  
emp.last_name AS Empleado,  
emp.job_id AS Puesto,  
emp.hire_date AS 'Fecha de contratación'  
FROM employees AS emp;
```

```
SELECT emp.employee_id AS 'Emp N',  
emp.last_name AS employee,  
emp.job_id AS jobs,  
emp.hire_date AS 'Fecha de contratación'  
FROM employees AS emp;  
go
```

|    | Emp N | employee  | jobs       | Fecha de contratación |
|----|-------|-----------|------------|-----------------------|
| 1  | 100   | King      | AD_PRES    | 2003-06-17            |
| 2  | 101   | Kochhar   | AD_VP      | 2005-09-21            |
| 3  | 102   | De Haan   | AD_VP      | 2001-01-13            |
| 4  | 103   | Hunold    | IT_PROG    | 2006-01-03            |
| 5  | 104   | Ernst     | IT_PROG    | 2007-05-21            |
| 6  | 105   | Austin    | IT_PROG    | 2005-06-25            |
| 7  | 106   | Pataballa | IT_PROG    | 2006-02-05            |
| 8  | 107   | Lorentz   | IT_PROG    | 2007-02-07            |
| 9  | 108   | Greenberg | FI_MGR     | 2002-08-17            |
| 10 | 109   | Faviet    | FI_ACCOUNT | 2002-08-16            |
| 11 | 110   | Chen      | FI_ACCOUNT | 2005-09-28            |
| 12 | 111   | Sciarra   | FI_ACCOUNT | 2005-09-30            |
| 13 | 112   | Uman      | FI_ACCOUNT | 2006-03-07            |
| 14 | 113   | Popp      | FI_ACCOUNT | 2007-12-07            |
| 15 | 114   | Raphaely  | PU_MAN     | 2002-12-07            |

2. Adicionalmente el departamento de Recursos Humanos requiere un reporte más sencillo, en el que se muestre los campos: last\_name y job.id en una sola y única columna (los datos deben estar separados por una coma) que tenga como alias Empleado y Puesto.

```
SELECT CONCAT(emp.last_name,',',emp.job_id) AS 'Empleado y Puesto'  
  
FROM employees AS emp;
```



```
SELECT CONCAT(emp.last_name,',',emp.job_id) AS 'Empleado y Puesto'
FROM employees AS emp;
go
```

| Results |                    | Messages |
|---------|--------------------|----------|
|         | Empleado y Puesto  |          |
| 1       | King,AD_PRES       |          |
| 2       | Kochhar,AD_VP      |          |
| 3       | De Haan,AD_VP      |          |
| 4       | Hunold,IT_PROG     |          |
| 5       | Ernst,IT_PROG      |          |
| 6       | Austin,IT_PROG     |          |
| 7       | Pataballa,IT_PROG  |          |
| 8       | Lorentz,IT_PROG    |          |
| 9       | Greenberg,FI_MGR   |          |
| 10      | Faviet,FI_ACCOUNT  |          |
| 11      | Chen,FI_ACCOUNT    |          |
| 12      | Sciarra,FI_ACCOUNT |          |
| 13      | Uman,FI_ACCOUNT    |          |
| 14      | Popp,FI_ACCOUNT    |          |
| 15      | Raphaely,PU_MAN    |          |

3. Finalmente a modo de práctica, realizar una consulta que muestre todos los campos de la tabla EMPLOYEES, en una sola y única columna, los datos deben estar separados por una coma y la columna debe tener como encabezado Los Empleados

```
SELECT CONCAT(emp.employee_id,',',
emp.first_name,',',
emp.last_name,',',
emp.email,',',
emp.phone_number,',',
emp.hire_date,',',
emp.job_id,',',
emp.salary,',',
emp.commission_pct,',',
emp.manager_id,',',
emp.department_id) AS 'Los empleados'
FROM employees AS emp;
```

```

SELECT CONCAT(emp.employee_id,',',
emp.first_name,',',
emp.last_name,',',
emp.email,',',
emp.phone_number,',',
emp.hire_date,',',
emp.job_id,',',
emp.salary,',',
emp.commission_pct,',',
emp.manager_id,',',
emp.department_id) AS 'Los empleados'
FROM employees AS emp;
go

```

Results

Messages

|    | Los empleados   |
|----|---|
| 1  | 100,Steven,King,SKING,515.123.4567,2003-06-17,AD_P...     |
| 2  | 101,Neena,Kochhar,NKOCHHAR,515.123.4568,2005-09-...       |
| 3  | 102,Lex,De Haan,LDEHAAN,515.123.4569,2001-01-13,A...      |
| 4  | 103,Alexander,Hunold,AHUNOLD,590.423.4567,2006-01-...     |
| 5  | 104,Bruce,Ernst,BERNST,590.423.4568,2007-05-21,IT_P...    |
| 6  | 105,David,Austin,DAUSTIN,590.423.4569,2005-06-25,IT_...   |
| 7  | 106,Valli,Pataballa,VPATABAL,590.423.4560,2006-02-05,I... |
| 8  | 107,Diana,Lorentz,DLORENTZ,590.423.5567,2007-02-07,...    |
| 9  | 108,Nancy,Greenberg,NGREENBE,515.124.4569,2002-08-...     |
| 10 | 109,Daniel,Faviet,DFAVIET,515.124.4169,2002-08-16,FI_...  |
| 11 | 110,John,Chen,JCHEN,515.124.4269,2005-09-28,FI_ACC...     |
| 12 | 111,Ismael,Sciarra,ISCIARRA,515.124.4369,2005-09-30,FI... |
| 13 | 112,Jose Manuel,Uman,JMURMAN,515.124.4469,2006-0...       |
| 14 | 113,Luis,Popp,LPOPP,515.124.4567,2007-12-07,FI_ACCO...    |
| 15 | 114,Den,Raphaely,DRAPHEAL,515.127.4561,2002-12-07,...     |

## 4. Actividad No 04 – Restricción y Ordenamiento

1. Debido a problemas con el presupuesto, el departamento de Recursos Humanos requiere un reporte que muestre los apellidos (last\_name) y salarios (salary) de todos los empleados que ganen más de \$ 12,000.

```
select last_name,salary from employees where salary >12000;
```

```
select last_name,salary from employees where salary > 12000;  
go
```

|   | last_name | salary   |
|---|-----------|----------|
| 1 | King      | 24000.00 |
| 2 | Kochhar   | 17000.00 |
| 3 | De Haan   | 17000.00 |
| 4 | Greenberg | 12008.00 |
| 5 | Russell   | 14000.00 |
| 6 | Partners  | 13500.00 |
| 7 | Hartstein | 13000.00 |
| 8 | Higgins   | 12008.00 |

2. Asimismo se requiere realizar una consulta que muestre los apellidos (last\_name) y el número de departamento (department\_id) para los empleados que tengan numero (employee\_id) 176.

```
select last_name,department_id from employees where employee_id >176;
```

```
select last_name,department_id from employees where employee_id > 176;  
go
```

|    | last_name  | department_id |
|----|------------|---------------|
| 1  | Livingston | 80            |
| 2  | Grant      | NULL          |
| 3  | Johnson    | 80            |
| 4  | Taylor     | 50            |
| 5  | Fleur      | 50            |
| 6  | Sullivan   | 50            |
| 7  | Geoni      | 50            |
| 8  | Sarchand   | 50            |
| 9  | Bull       | 50            |
| 10 | Dellinger  | 50            |
| 11 | Cabrio     | 50            |
| 12 | Chung      | 50            |
| 13 | Dilly      | 50            |
| 14 | Gates      | 50            |
| 15 | Perkins    | 50            |

3. El departamento de Recursos Humanos necesita determinar los mayores y menores sueldos, modificar la consulta del ítem 4.1. para mostrar el apellido y salario de cada empleado cuyo sueldo no esté en el rango de \$ 5,000 a \$ 12,000.

`select last_name,job_id,salary as Sal from employees where salary < 5000 and salary > 12000;`

```
select last_name,job_id,salary as Sal from employees where salary > 5000 and salary < 12000;  
go
```

| Results |           | Messages   |          |
|---------|-----------|------------|----------|
|         | last_name | job_id     | Sal      |
| 1       | Hunold    | IT_PROG    | 9000.00  |
| 2       | Ernst     | IT_PROG    | 6000.00  |
| 3       | Faviet    | FI_ACCOUNT | 9000.00  |
| 4       | Chen      | FI_ACCOUNT | 8200.00  |
| 5       | Sciarra   | FI_ACCOUNT | 7700.00  |
| 6       | Urman     | FI_ACCOUNT | 7800.00  |
| 7       | Popp      | FI_ACCOUNT | 6900.00  |
| 8       | Raphaely  | PU_MAN     | 11000.00 |
| 9       | Weiss     | ST_MAN     | 8000.00  |
| 10      | Fripp     | ST_MAN     | 8200.00  |
| 11      | Kaufling  | ST_MAN     | 7900.00  |
| 12      | Vollman   | ST_MAN     | 6500.00  |
| 13      | Mourgos   | ST_MAN     | 5800.00  |
| 14      | Cambraut  | SA_MAN     | 11000.00 |
| 15      | Zlotkey   | SA_MAN     | 10500.00 |

4. Crear un reporte que muestre los apellidos (last\_name), puesto (job\_id) y fecha de contratación (hire\_date), de los empleados que apellidan 'Matos' y 'Taylor', asimismo presentar el reporte ordenado ascendentemente por fecha de contratación.

`select last_name,job_id,hire_date from employees where last_name = 'Matos' or last_name = 'Taylor' order by hire_date asc;`

```
select last_name,job_id,hire_date from employees where  
last_name = 'Matos' or last_name = 'Taylor' order by hire_date asc;  
go
```

| Results |           | Messages |            |
|---------|-----------|----------|------------|
|         | last_name | job_id   | hire_date  |
| 1       | Taylor    | SH_CLERK | 2006-01-24 |
| 2       | Matos     | ST_CLERK | 2006-03-15 |
| 3       | Taylor    | SA_REP   | 2006-03-24 |

5. Mostrar los apellidos (last\_name) y número de departamento (departamento\_id) de todos los empleados que pertenezcan a los departamentos 20 o 50 en orden alfabético ascendente por el apellido.

`select last_name,department_id from employees where department_id = 20 or department_id = 50 order by last_name asc;`

```
select last_name,department_id from employees where
department_id = 20 or department_id = 50 order by last_name asc;
go
```

|    | last_name | department_id |
|----|-----------|---------------|
| 1  | Atkinson  | 50            |
| 2  | Bell      | 50            |
| 3  | Bissot    | 50            |
| 4  | Bull      | 50            |
| 5  | Cabrio    | 50            |
| 6  | Chung     | 50            |
| 7  | Davies    | 50            |
| 8  | Dellinger | 50            |
| 9  | Dilly     | 50            |
| 10 | Everett   | 50            |
| 11 | Fay       | 20            |
| 12 | Feeney    | 50            |
| 13 | Fleur     | 50            |
| 14 | Fripp     | 50            |
| 15 | Gates     | 50            |

6. Modificar el reporte del ítem 4.1. para mostrar los apellidos y salarios de los empleados que tengan un salario entre los \$ 5,000 a \$ 12,000 y pertenezcan a los números de departamento 20 o 50. Asimismo etiquetar las cabeceras de los resultados con los alias Empleado y Salario Mensual respectivamente.

```
select last_name 'Empleado',salary 'Salario Mensual' from employees where salary >5000 and salary <12000 and (department_id = 20 or department_id = 50);
```

```
select last_name 'Empleado', salary 'Salario Mensual' from employees where
salary > 5000 and salary < 12000 and (department_id = 20 or department_id = 50);
go
```

|   | Empleado | Salario Mensual |
|---|----------|-----------------|
| 1 | Weiss    | 8000.00         |
| 2 | Fripp    | 8200.00         |
| 3 | Kaufling | 7900.00         |
| 4 | Vollman  | 6500.00         |
| 5 | Mourgos  | 5800.00         |
| 6 | Fay      | 6000.00         |

7. El departamento de Recursos Humanos necesita un listado de apellidos (last\_name) y fecha de contratación (hire\_date) de todos los empleados que fueron contratados el año 1994.

```
select last_name,hire_date from employees where hire_date between '19940101' and '19941231';
```

```
select last_name,hire_date from employees where hire_date between '19940101' and '19941231';
go
```

| Results   |           | Messages |
|-----------|-----------|----------|
| last_name | hire_date |          |

8. Crear un reporte que muestre los apellidos (last\_name) y puesto (job\_id) de todos los empleados que no tengan un administrador (manager).

```
select last_name,job_id from employees where manager_id is null;
```

```
select last_name,job_id from employees where manager_id is null;
go
```

| Results |           | Messages |
|---------|-----------|----------|
|         | last_name | job_id   |
| 1       | King      | AD_PRES  |

9. Crear un reporte para mostrar los apellidos (last\_name), salario (salary) y % de comisión (commission\_pct). Ordenar los datos por salario y comisión de manera descendente, utilizar la opción numérica de la cláusula ORDER BY.

```
select last_name,salary,commission_pct from employees order by salary desc,commission_pct desc;
```

```
select last_name,salary,commission_pct from employees order by salary desc,commission_pct desc;
go
```

| Results |           | Messages |                |
|---------|-----------|----------|----------------|
|         | last_name | salary   | commission_pct |
| 1       | King      | 24000.00 | NULL           |
| 2       | Kochhar   | 17000.00 | NULL           |
| 3       | De Haan   | 17000.00 | NULL           |
| 4       | Russell   | 14000.00 | 0.40           |
| 5       | Partners  | 13500.00 | 0.30           |
| 6       | Hartstein | 13000.00 | NULL           |
| 7       | Higgins   | 12008.00 | NULL           |
| 8       | Greenberg | 12008.00 | NULL           |
| 9       | Errazuriz | 12000.00 | 0.30           |
| 10      | Ozer      | 11500.00 | 0.25           |
| 11      | Cambraut  | 11000.00 | 0.30           |
| 12      | Abel      | 11000.00 | 0.30           |
| 13      | Raphaely  | 11000.00 | NULL           |
| 14      | Vishney   | 10500.00 | 0.25           |
| 15      | Zlotkey   | 10500.00 | 0.20           |

10. El personal del departamento de Recursos Humanos desea tener mayor flexibilidad con los reportes hechos. Por ejemplo se requiere un reporte de los apellidos (last\_name) y salarios (salary) de todos los empleados que tengan un salario mayor a un monto que el personal de Recursos Humanos ingresará. Probar con el valor \$ 12,000.

```
declare @salario as decimal(9,2); set @salario = 12000; select last_name,salary from employees where salary > @salario;
```

```
declare @salario as decimal(9,2); set @salario = 12000; select last_name, salary from employees where salary > @salario;
```

| Results |           | Messages |
|---------|-----------|----------|
|         | last_name | salary   |
| 1       | King      | 24000.00 |
| 2       | Kochhar   | 17000.00 |
| 3       | De Haan   | 17000.00 |
| 4       | Greenberg | 12008.00 |
| 5       | Russell   | 14000.00 |
| 6       | Partners  | 13500.00 |
| 7       | Hartstein | 13000.00 |
| 8       | Higgins   | 12008.00 |

11. El departamento de Recursos Humanos requiere extraer reporte basados en el Administrador (manager\_id). Se requiere crear una consulta que pregunte al usuario por el Administrador (manager\_id) y genere un reporte con los números de empleado (employee\_id), apellidos (last\_name), salarios (salary) y numero de departamento de los empleados que este Administrador tiene a su cargo. Adicionalmente también se desea tener la habilidad de ordenar este reporte en base a una determinada columna. Probar con los siguientes valores:

Administrador (manager\_id) = 103, ordenado por Apellido (last\_name)

Administrador (manager\_id) = 201, ordenado por Salario (salary)

Administrador (manager\_id) = 124, ordenado por No de Empleado (employee\_id)

```
declare @gerente as int;
set @gerente = 103;
select employee_id,last_name,salary,department_id from employees where manager_id = @gerente order by last_name;
set @gerente = 201;
select employee_id,last_name,salary,department_id from employees where manager_id = @gerente order by salary;
set @gerente = 124;
select employee_id,last_name,salary,department_id from employees where manager_id = @gerente order by employee_id;
go
```

```

declare @gerente as int;
set @gerente = 103;
select employee_id,last_name,salary,department_id from employees where manager_id = @gerente order by last_name;
set @gerente = 201;
select employee_id,last_name,salary,department_id from employees where manager_id = @gerente order by salary;
set @gerente = 124;
select employee_id,last_name,salary,department_id from employees where manager_id = @gerente order by employee_id;
go

```

|   | employee_id | last_name | salary  | department_id |
|---|-------------|-----------|---------|---------------|
| 1 | 105         | Austin    | 4800.00 | 60            |
| 2 | 104         | Emst      | 6000.00 | 60            |
| 3 | 107         | Lorentz   | 4200.00 | 60            |
| 4 | 106         | Pataballa | 4800.00 | 60            |

|   | employee_id | last_name | salary  | department_id |
|---|-------------|-----------|---------|---------------|
| 1 | 202         | Fay       | 6000.00 | 20            |

|   | employee_id | last_name | salary  | department_id |
|---|-------------|-----------|---------|---------------|
| 1 | 141         | Rajs      | 3500.00 | 50            |
| 2 | 142         | Davies    | 3100.00 | 50            |
| 3 | 143         | Matos     | 2600.00 | 50            |
| 4 | 144         | Vargas    | 2500.00 | 50            |
| 5 | 196         | Walsh     | 3100.00 | 50            |
| 6 | 197         | Feeney    | 3000.00 | 50            |
| 7 | 198         | OConnell  | 2600.00 | 50            |
| 8 | 199         | Grant     | 2600.00 | 50            |

12. Generar un listado de apellidos (last\_name) de todos los empleados que tengan la letra 'a' en la tercera letra de su apellido.

```

select last_name from employees where SUBSTRING(last_name,3,1) = 'a';
go

```

```

select last_name from employees where SUBSTRING(last_name,3,1) = 'a';
go

```

|   | last_name |
|---|-----------|
| 1 | Grant     |
| 2 | Grant     |
| 3 | Whalen    |

13. Mostrar los apellidos (last\_name) de todos los empleados que tengan tanto la letra 'a' como la letra 'e' en su apellido.

```

select last_name from employees where SUBSTRING(last_name,3,1) = 'a' or SUBSTRING(last_name,3,1) = 'e';
go

```



```
select last_name from employees where SUBSTRING(last_name,3,1) = 'a' or SUBSTRING(last_name,3,1) = 'e';
go
```

| Results |           | Messages |
|---------|-----------|----------|
|         | last_name |          |
| 1       | Greenberg |          |
| 2       | Chen      |          |
| 3       | Gee       |          |
| 4       | McEwen    |          |
| 5       | Greene    |          |
| 6       | Lee       |          |
| 7       | Ozer      |          |
| 8       | Abel      |          |
| 9       | Grant     |          |
| 10      | Fleaur    |          |

14. Mostrar los apellidos (last\_name), puestos (job\_id) y salario (salary) de todos los empleados que sean Representantes de Ventas (SA\_REP) o Responsables de Inventario (ST\_CLERK) y cuyos salarios no sean iguales a \$ 2,500, \$ 3,500 o \$ 7,000.

```
select last_name,job_id,salary from employees where (job_id = 'SA_REP' or job_id = 'ST_CLERK')
and (salary = 2500 or salary = 3500 or salary = 7000);
go
```

```
select last_name,job_id,salary from employees where
(job_id = 'SA_REP' or job_id = 'ST_CLERK') and (salary = 2500 or salary = 3500 or salary = 7000);
go
```

Results

Messages

|   | last_name | job_id   | salary  |
|---|-----------|----------|---------|
| 1 | Marlow    | ST_CLERK | 2500.00 |
| 2 | Patel     | ST_CLERK | 2500.00 |
| 3 | Rajs      | ST_CLERK | 3500.00 |
| 4 | Vargas    | ST_CLERK | 2500.00 |
| 5 | Tuvault   | SA_REP   | 7000.00 |
| 6 | Sewall    | SA_REP   | 7000.00 |
| 7 | Grant     | SA_REP   | 7000.00 |

15. Modificar el reporte del ítem 4.6 y mostrar adicionalmente los datos de comisión (commission\_pct) de todos los empleados que solamente el 20 % de comisión.

```
select last_name 'Empleado',salary 'Salario Mensual',commission_pct from employees where salary >5000 and salary <12000 and (department_id = 20 or department_id = 50) and commission_pct = 0.20;
go
```

```
select last_name 'Empleado', salary 'Salario Mensual', commission_pct from employees where  
salary > 5000 and salary < 12000 and (department_id = 20 or department_id = 50) and commission_pct = 0.20;  
go
```

| Results  |  |  | Messages        |  |  |
|----------|--|--|-----------------|--|--|
| Empleado |  |  | Salario Mensual |  |  |
|          |  |  | commission_pct  |  |  |
|          |  |  |                 |  |  |

## 5. Actividad No 05 – Funciones

1. Se requiere realizar una consulta que visualice la fecha del sistema.

```
SELECT CONVERT (date, SYSDATETIME())
,CONVERT (date, SYSDATETIMEOFFSET())
,CONVERT (date, SYSUTCDATETIME())
,CONVERT (date, CURRENT_TIMESTAMP)
,CONVERT (date, GETDATE())
,CONVERT (date, GETUTCDATE());
```

```
]SELECT CONVERT (date, SYSDATETIME())
,CONVERT (date, SYSDATETIMEOFFSET())
,CONVERT (date, SYSUTCDATETIME())
,CONVERT (date, CURRENT_TIMESTAMP)
,CONVERT (date, GETDATE())
,CONVERT (date, GETUTCDATE());
go
```

| Results |                  | Messages         |                  |                  |                  |                  |
|---------|------------------|------------------|------------------|------------------|------------------|------------------|
|         | (No column name) | (No column name) | (No column name) | (No column name) | (No column name) | (No column name) |
| 1       | 2018-09-29       | 2018-09-29       | 2018-09-29       | 2018-09-29       | 2018-09-29       | 2018-09-29       |

2. El departamento de Recursos Humanos necesita un reporte de todos los empleados que muestre el No de Empleado, Apellidos, Salario y una columna más con el cálculo del salario incrementado en 15.5 % (expresado solo en enteros) esta columna debe etiquetarse Nuevo Salario

```
SELECT employee_id,last_name,salary,salary*0.155 as newsalary FROM employees
```

```
SELECT employee_id, last_name, salary, salary*0.155 as newsalary FROM employees;
go
```

| Results |             | Messages  |          |            |  |
|---------|-------------|-----------|----------|------------|--|
|         | employee_id | last_name | salary   | newsalary  |  |
| 1       | 100         | King      | 24000.00 | 3720.00000 |  |
| 2       | 101         | Kochhar   | 17000.00 | 2635.00000 |  |
| 3       | 102         | De Haan   | 17000.00 | 2635.00000 |  |
| 4       | 103         | Hunold    | 9000.00  | 1395.00000 |  |
| 5       | 104         | Ernst     | 6000.00  | 930.00000  |  |
| 6       | 105         | Austin    | 4800.00  | 744.00000  |  |
| 7       | 106         | Pataballa | 4800.00  | 744.00000  |  |
| 8       | 107         | Lorentz   | 4200.00  | 651.00000  |  |
| 9       | 108         | Greenberg | 12008.00 | 1861.24000 |  |
| 10      | 109         | Faviet    | 9000.00  | 1395.00000 |  |
| 11      | 110         | Chen      | 8200.00  | 1271.00000 |  |
| 12      | 111         | Sciarra   | 7700.00  | 1193.50000 |  |
| 13      | 112         | Urman     | 7800.00  | 1209.00000 |  |
| 14      | 113         | Popp      | 6900.00  | 1069.50000 |  |
| 15      | 114         | Raphaely  | 11000.00 | 1705.00000 |  |

3. Modificar la consulta anterior y adicionar una columna que muestre el resultado de la resta entre el antiguo salario y el nuevo salario. Etiquetar esta columna como Incremento.

SELECT employee\_id,last\_name,salary,salary\*0.155 as newsalary,salary-(salary\*0.155) as incremento FROM employees

```
SELECT employee_id,last_name,salary,salary*0.155 as newsalary,salary-(salary*0.155) as incremento FROM employees
go
```

| Results |             | Messages  |          |            |             |
|---------|-------------|-----------|----------|------------|-------------|
|         | employee_id | last_name | salary   | newsalary  | incremento  |
| 1       | 100         | King      | 24000.00 | 3720.00000 | 20280.00000 |
| 2       | 101         | Kochhar   | 17000.00 | 2635.00000 | 14365.00000 |
| 3       | 102         | De Haan   | 17000.00 | 2635.00000 | 14365.00000 |
| 4       | 103         | Hunold    | 9000.00  | 1395.00000 | 7605.00000  |
| 5       | 104         | Ernst     | 6000.00  | 930.00000  | 5070.00000  |
| 6       | 105         | Austin    | 4800.00  | 744.00000  | 4056.00000  |
| 7       | 106         | Pataballa | 4800.00  | 744.00000  | 4056.00000  |
| 8       | 107         | Lorentz   | 4200.00  | 651.00000  | 3549.00000  |
| 9       | 108         | Greenberg | 12008.00 | 1861.24000 | 10146.76000 |
| 10      | 109         | Faviet    | 9000.00  | 1395.00000 | 7605.00000  |
| 11      | 110         | Chen      | 8200.00  | 1271.00000 | 6929.00000  |
| 12      | 111         | Sciarra   | 7700.00  | 1193.50000 | 6506.50000  |
| 13      | 112         | Uman      | 7800.00  | 1209.00000 | 6591.00000  |
| 14      | 113         | Popp      | 6900.00  | 1069.50000 | 5830.50000  |
| 15      | 114         | Raphaely  | 11000.00 | 1705.00000 | 9295.00000  |

4. Crear un reporte que muestre los Apellidos (con la primera letra en Mayúsculas y las demás en Minúsculas) y la longitud de los apellidos (colocar alias Longitud), para todos aquellos empleados quienes sus apellidos empiecen con las letras 'J', 'A' y 'M'. Ordenar los resultados por la columna Apellido.

```
select UPPER(last_name) "Apellido", (LOWER(first_name)) "Longitud"
from employees
where last_name like 'A %'
or last_name like 'J %'
or last_name like 'M %' order by last_name asc;
```

```
select UPPER(last_name) 'Apellido', (LOWER(first_name)) 'Longitud' from employees
where last_name like 'A%'
or last_name like 'J%'
or last_name like 'M%' order by last_name asc;
go
```

| Results |             | Messages |
|---------|-------------|----------|
|         | Apellido    | Longitud |
| 1       | ABEL        | ellen    |
| 2       | ANDE        | sundar   |
| 3       | ATKINSON    | mozhe    |
| 4       | AUSTIN      | david    |
| 5       | JOHNSON     | charles  |
| 6       | JONES       | vance    |
| 7       | MALLIN      | jason    |
| 8       | MARKLE      | steven   |
| 9       | MARLOW      | james    |
| 10      | MARVINS     | mattea   |
| 11      | MATOS       | randall  |
| 12      | MAVRIS      | susan    |
| 13      | MCCAIN      | samuel   |
| 14      | MCEWEN      | allan    |
| 15      | MIKKILINENI | irene    |

5. Modificar la consulta anterior a fin de que consulte primero al usuario con que letra empieza el apellido a buscar. Considerar que no importa si la letra esta mayúscula o minúscula de igual manera debe mostrar los resultados.

```
select initcap(FIRST_NAME) as "name", length(first_name) as "Length" from employees where
upper(substr(first_name,1,1))=upper('&Inicial') order by first_name;
```

```

-- Drop la función si existe
IF OBJECT_ID('dbo.InitCap') IS NOT NULL
    DROP FUNCTION dbo.InitCap;
GO
-- Implementando la función de Oracle INITCAP en SQL Server
CREATE FUNCTION dbo.InitCap (@inStr VARCHAR(8000))
RETURNS VARCHAR(8000)
AS
BEGIN
    DECLARE @outStr VARCHAR(8000) = LOWER(@inStr),
            @char CHAR(1),
            @alphanum BIT = 0,
            @len INT = LEN(@inStr),
            @pos INT = 1;
    -- Iterar entre todos los caracteres en la cadena de entrada
    WHILE @pos <= @len BEGIN
        -- Obtener el siguiente caracter
        SET @char = SUBSTRING(@inStr, @pos, 1);
        -- Si la posición del caracter es la 1ª, o el caracter previo no es alfanumérico
        -- convierte el caracter actual a mayúscula
        IF @pos = 1 OR @alphanum = 0
            SET @outStr = STUFF(@outStr, @pos, 1, UPPER(@char));
        SET @pos = @pos + 1;
        -- Define si el caracter actual es non-alfanumérico
        IF ASCII(@char) <= 47 OR (ASCII(@char) BETWEEN 58 AND 64) OR
            (ASCII(@char) BETWEEN 91 AND 96) OR (ASCII(@char) BETWEEN 123 AND 126)
            SET @alphanum = 0;
        ELSE
            SET @alphanum = 1;
    END
    RETURN @outStr;
END
GO

select dbo.InitCap(FIRST_NAME) as
'name', len(first_name) as
'Length' from employees where
upper(SUBSTRING(first_name, 1, 1)) = upper('&Inicial')
order by first_name;

```

| Results |  | Messages |  |
|---------|--|----------|--|
| name    |  | Length   |  |
|         |  |          |  |

6. El departamento de Recursos Humanos la duración o tiempo de permanencia de cada empleado, mostrar el Apellido y el calculo del número de meses entre la fecha de hoy y la fecha en que fue contratado el empleado, Etiquetar la columna como Meses Trabajados, ordenar los resultados por el resultado de los números de meses, Redondear el número de meses al entero más cercano.

```

SELECT LAST_NAME, ROUND(MONTHS_BETWEEN(SYSDATE,HIRE_DATE),0) "MONTHS_WORKED"
from employees order by MONTHS_BETWEEN( HIRE_DATE, SYSDATE);

```

```

GO
CREATE FUNCTION MONTHS_BETWEEN (@date1 DATETIME, @date2 DATETIME)
RETURNS FLOAT AS
BEGIN
    DECLARE @months FLOAT = DATEDIFF(month, @date2, @date1);
    -- Both dates does not point to the same day of month
    IF DAY(@date1) <> DAY(@date2) AND
        -- Both dates does not point to the last day of month
        (MONTH(@date1) = MONTH(@date1 + 1) OR MONTH(@date2) = MONTH(@date2 + 1))
    BEGIN
        -- Correct to include full months only and calculate fraction
        IF DAY(@date1) < DAY(@date2)
            SET @months = @months + CONVERT(FLOAT, 31 - DAY(@date2) + DAY(@date1)) / 31 - 1;
        ELSE
            SET @months = @months + CONVERT(FLOAT, DAY(@date1) - DAY(@date2)) / 31;
        END
    END
    RETURN @months;
END;
GO
go
SELECT last_name, ROUND(dbo.MONTHS_BETWEEN(SYSDATETIME(), HIRE_DATE),0) 'MONTHS_WORK'
from employees order by dbo.MONTHS_BETWEEN(HIRE_DATE, SYSDATETIME());
go

```

|    | last_name | MONTHS_WORK |
|----|-----------|-------------|
| 1  | De Haan   | 213         |
| 2  | Mavris    | 196         |
| 3  | Baer      | 196         |
| 4  | Higgins   | 196         |
| 5  | Gietz     | 196         |
| 6  | Faviet    | 193         |
| 7  | Greenberg | 193         |
| 8  | Raphaely  | 190         |
| 9  | Kaufling  | 185         |
| 10 | Khoo      | 184         |
| 11 | King      | 183         |
| 12 | Ladwig    | 182         |
| 13 | Whalen    | 180         |
| 14 | Rajs      | 179         |
| 15 | Sarchand  | 176         |

7. Crear una consulta que devuelva los Apellidos y Salarios de todos los empleados, Formatear la columna salario para que muestre 15 caracteres, completar con el símbolo '\$' los espacios previos al valor de la columna salario, ejemplo: \$\$\$\$\$\$\$\$\$\$10000. Etiquetar esta columna como Salario.

```

CREATE FUNCTION LPAD
(
    @string VARCHAR(MAX),
    @length INT,
    @pad CHAR
)
RETURNS VARCHAR(MAX)
AS
BEGIN
    RETURN REPLICATE(@pad, @length - LEN(@string)) + @string;
END

```

```
GO
SELECT dbo.LPAD(salary, 15, '$') VALUE
FROM employees;
```

```
CREATE FUNCTION LPAD
(
    @string VARCHAR(MAX),
    @length INT,
    @pad CHAR
)
RETURNS VARCHAR(MAX)
AS
BEGIN
    RETURN REPLICATE(@pad, @length - LEN(@string)) + @string;
END
GO
SELECT dbo.LPAD(salary, 15, '$') VALUE
FROM employees;
```

| Results |                        | Messages |
|---------|------------------------|----------|
|         | VALUE                  |          |
| 1       | \$\$\$\$\$\$\$24000.00 |          |
| 2       | \$\$\$\$\$\$\$17000.00 |          |
| 3       | \$\$\$\$\$\$\$17000.00 |          |
| 4       | \$\$\$\$\$\$\$9000.00  |          |
| 5       | \$\$\$\$\$\$\$6000.00  |          |
| 6       | \$\$\$\$\$\$\$4800.00  |          |
| 7       | \$\$\$\$\$\$\$4800.00  |          |
| 8       | \$\$\$\$\$\$\$4200.00  |          |
| 9       | \$\$\$\$\$\$\$12008.00 |          |
| 10      | \$\$\$\$\$\$\$9000.00  |          |
| 11      | \$\$\$\$\$\$\$8200.00  |          |
| 12      | \$\$\$\$\$\$\$7700.00  |          |
| 13      | \$\$\$\$\$\$\$7800.00  |          |
| 14      | \$\$\$\$\$\$\$6900.00  |          |
| 15      | \$\$\$\$\$\$\$11000.00 |          |

8. Crear una consulta que muestre en una única columna los primeros 8 caracteres del apellido de los empleados e indique sus salarios representados por asteriscos (\*), cada asterisco representa el valor 1000. Ordenar el listado por el salario de los empleados. Asimismo Etiquetar la columna como 'Empleados y sus Salarios'.
9. Finalmente crear una consulta que muestre los Apellidos de los empleados y el No de Semanas Empleado hasta la actualidad para todos los empleados del departamento No 90, truncar el número de semanas a sin decimales. Ordenar el resultado por el No de Semanas y etiquetar la columna como tenencia.



select last\_name, TRUNC(((SYSDATE-hire\_date)/7),0) as TENURE from employees where department\_id=90 ORDER BY hire\_date DESC;

```
CREATE FUNCTION dbo.trunc (@input datetime)
    RETURNS datetime
AS
BEGIN
    DECLARE @fecha datetime,
            @fechastring varchar(10)
    SET @fechastring = CONVERT(varchar(10),@input, 103)
    SET @fecha = CONVERT(datetime, @fechastring, 103)
    RETURN @fecha
END
GO
select last_name, dbo.trunc((((CONVERT (date, SYSDATETIME()) - hire_date)/7),0) as
TENURE from employees where department_id=90 ORDER BY hire_date DESC;
```

## 6. Actividad No 06 – Funciones de Conversión

1. Crear un reporte que muestre lo siguiente por cada empleado.  
(Apellido del empleado) gana (Salario) pero quisiera (3 veces Salario).  
Etiquetar la columna como Sueldos Soñados.

```
select 'Sueldos Soñados'=(last_name + ' gana ' + Cast(salary as varchar(18)) + ' pero  
quisiera ' + Cast((salary * 3) as varchar(18)))  
from dbo.employees  
go
```

```
go  
select 'Sueldos Soñados'=(last_name + ' gana ' + Cast(salary as varchar(18)) +  
'pero quisiera' + Cast((salary * 3) as varchar(18)))  
from employees  
go
```

|    | Sueldos Soñados                            |
|----|--|
| 1  | Kinggana24000.00pero quisiera72000.00      |
| 2  | Kochhargana17000.00pero quisiera51000.00   |
| 3  | De Haangana17000.00pero quisiera51000.00   |
| 4  | Hunoldgana9000.00pero quisiera27000.00     |
| 5  | Emstgana6000.00pero quisiera18000.00       |
| 6  | Austingana4800.00pero quisiera14400.00     |
| 7  | Pataballagana4800.00pero quisiera14400.00  |
| 8  | Lorentzgana4200.00pero quisiera12600.00    |
| 9  | Greenberggana12008.00pero quisiera36024.00 |
| 10 | Favietgana9000.00pero quisiera27000.00     |
| 11 | Chengana8200.00pero quisiera24600.00       |
| 12 | Sciaragana7700.00pero quisiera23100.00     |
| 13 | Umangana7800.00pero quisiera23400.00       |
| 14 | Poppgana6900.00pero quisiera20700.00       |
| 15 | Raphaelygana11000.00pero quisiera33000.00  |

2. Realizar una consulta que muestre el Apellido del empleado, fecha de contratación y la Fecha de Revisión del Salario, la cual es el primer Lunes después de cada seis meses de servicio, etiquetar la columna como Revisión, asimismo el formato de esta fecha debe ser similar al siguiente:

Lunes, el veintiuno de julio, 2003

```
select last_name, hire_date as Revision from employees  
where hire_date between '2003-06-17' and '2005-09-21';  
go
```

```
select last_name, hire_date as Revision from employees
where hire_date between '2003-06-17' and '2005-09-21';
go
```

| Results |           | Messages   |
|---------|-----------|------------|
|         | last_name | Revision   |
| 1       | King      | 2003-06-17 |
| 2       | Kochhar   | 2005-09-21 |
| 3       | Austin    | 2005-06-25 |
| 4       | Tobias    | 2005-07-24 |
| 5       | Weiss     | 2004-07-18 |
| 6       | Fripp     | 2005-04-10 |
| 7       | Nayer     | 2005-07-16 |
| 8       | Bissot    | 2005-08-20 |
| 9       | Marlow    | 2005-02-16 |
| 10      | Mallin    | 2004-06-14 |
| 11      | Ladwig    | 2003-07-14 |
| 12      | Rajs      | 2003-10-17 |
| 13      | Davies    | 2005-01-29 |
| 14      | Russell   | 2004-10-01 |
| 15      | Partners  | 2005-01-05 |

3. Mostrar un reporte que tenga los Apellidos, Fecha de Contratación y el Día de Inicio de cada empleado (Lunes, Martes, etc. . . ), etiquetar la última columna como Día. Ordenar los resultados por el Día de Inicio empezando por Lunes.

```
select e.last_name, e.hire_date, DateName(WEEKDAY, jh.START_DATE)as 'Dia'
from dbo.employees as e inner join dbo.job_history as jh on
e.employee_id=jh.employee_id
go
```

```
select e.last_name, e.hire_date, DateName(WEEKDAY, jh.START_DATE)as 'Dia'
from employees as e inner join job_history as jh on
e.employee_id=jh.employee_id
go
```

|    | last_name | hire_date  | Dia     |
|----|-----------|------------|---------|
| 1  | Kochhar   | 2005-09-21 | Domingo |
| 2  | Kochhar   | 2005-09-21 | Domingo |
| 3  | De Haan   | 2001-01-13 | Sábado  |
| 4  | Raphaely  | 2002-12-07 | Viernes |
| 5  | Kaufling  | 2003-05-01 | Lunes   |
| 6  | Taylor    | 2006-03-24 | Viernes |
| 7  | Taylor    | 2006-03-24 | Lunes   |
| 8  | Whalen    | 2003-09-17 | Domingo |
| 9  | Whalen    | 2003-09-17 | Lunes   |
| 10 | Hartstein | 2004-02-17 | Martes  |

4. Crear un listado que muestre los Apellidos de los empleados y sus Montos de Comisión, en caso no tenga comisión deberá mostrar el texto 'Sin Comisión', etiquetar esta ultima columna como Comisión.

Sintaxis y demostracion

```
select last_name as 'Apellidos', 'Comision'='Sin Comision' from dbo.employees where
commission_pct != 0 UNION select last_name as 'Apellidos', 'Comision'= Cast((salary * commission_pct)
as varchar(20)) from dbo.employees where commission_pct >0
```

|    | Apellidos | Comision     |
|----|-----------|--------------|
| 1  | Abel      | 3300.0000    |
| 2  | Abel      | Sin Comision |
| 3  | Ande      | 640.0000     |
| 4  | Ande      | Sin Comision |
| 5  | Banda     | 620.0000     |
| 6  | Banda     | Sin Comision |
| 7  | Bates     | 1095.0000    |
| 8  | Bates     | Sin Comision |
| 9  | Bernstein | 2375.0000    |
| 10 | Bernstein | Sin Comision |

5. Rescribir la consulta anterior utilizando la función CASE.

Sintaxis y demostracion

```
]select e.last_name as 'Apellidos', j.job_title, case when j.job_id = 'AD PRES' THEN 'A'
when j.job_id = 'ST_MAN' THEN 'B' when j.job_id = 'IT_PROG' THEN 'C' when j.job_id = 'SA REP'
THEN 'D' else '0' END as 'Grados' from dbo.employees as e inner join dbo.jobs
as j on e.job_id=j.job_id
```

|    | Apellidos | job_title                     | Grados |
|----|-----------|-------------------------------|--------|
| 1  | King      | President                     | 0      |
| 2  | Kochhar   | Administration Vice President | 0      |
| 3  | De Haan   | Administration Vice President | 0      |
| 4  | Hunold    | Programmer                    | C      |
| 5  | Emst      | Programmer                    | C      |
| 6  | Austin    | Programmer                    | C      |
| 7  | Pataballa | Programmer                    | C      |
| 8  | Lorentz   | Programmer                    | C      |
| 9  | Greenberg | Finance Manager               | 0      |
| 10 | Faviet    | Accountant                    | 0      |

## 7. Actividad No 07 – Funciones de Agrupación

1. El departamento de Recursos Humanos requiere un reporte que muestre el máximo, el mínimo, la suma y el promedio de los salarios de todos los empleados, Etiquetar esta columnas como Máximo, Mínimo, Suma y Promedio respectivamente, Redondear estos valores a enteros sin decimales.

Sintaxis y demostracion

```
select round (max(salary),0) as "Maximo", round (min(salary),0) as  
"Minimo", round(sum(salary),0)as "Sumatoria",round (avg(salary),0)as "Promedio" from employees;
```

|   | Maximo   | Minimo  | Sumatoria | Promedio    |
|---|----------|---------|-----------|-------------|
| 1 | 24000.00 | 2100.00 | 699716.00 | 6479.000000 |

2. Modificar la consulta anterior para mostrar el máximo, mínimo, suma y promedio de los salarios por cada Puesto de trabajo.
3. Realizar un reporte que muestre la cantidad de empleados por Puesto de trabajo. Con la opción de que el usuario pueda ingresar todos los puestos o uno solo.

Sintaxis y demostracion

```
select count (*) from employees group by job_id;
```

|    | (No column name) |
|----|------------------|
| 1  | 2                |
| 2  | 1                |
| 3  | 1                |
| 4  | 1                |
| 5  | 2                |
| 6  | 5                |
| 7  | 1                |
| 8  | 1                |
| 9  | 5                |
| 10 | 1                |
| 11 | 1                |
| 12 | 1                |
| 13 | 5                |
| 14 | 1                |
| 15 | 5                |
| 16 | 30               |
| 17 | 20               |
| 18 | 20               |
| 19 | 5                |

4. Determinar el número de Administradores o Supervisores utilizar la columna manager\_id para esto. Etiquetar la columna como No de Administradores

Sintaxis y demostracion

```
select count (distinct manager_id) as "Numero de Administradores" from employees;
```

|   | Numero de Administradores |
|---|---------------------------|
| 1 | 18                        |

5. Encontrar la diferencia entre el máximo y mínimo salario de los empleados. Etiquetar la columna como Diferencia

Sintaxis y demostracion

```
select (max(salary)-min(salary)) as "Diferencia" from employees;
```

|   | Diferencia |
|---|------------|
| 1 | 21900.00   |

6. Crear un reporte que muestre los No de Administradores (manager\_id) y el salario de su empleado peor pagado. Excluir a los empleados cuyo Administrador no se conozca. Excluir asimismo cualquier grupo cuyo salario mínimo sea \$6000 o menos. Ordenar los resultados por el mínimo salario en forma descendente.

Sintaxis y demostracion

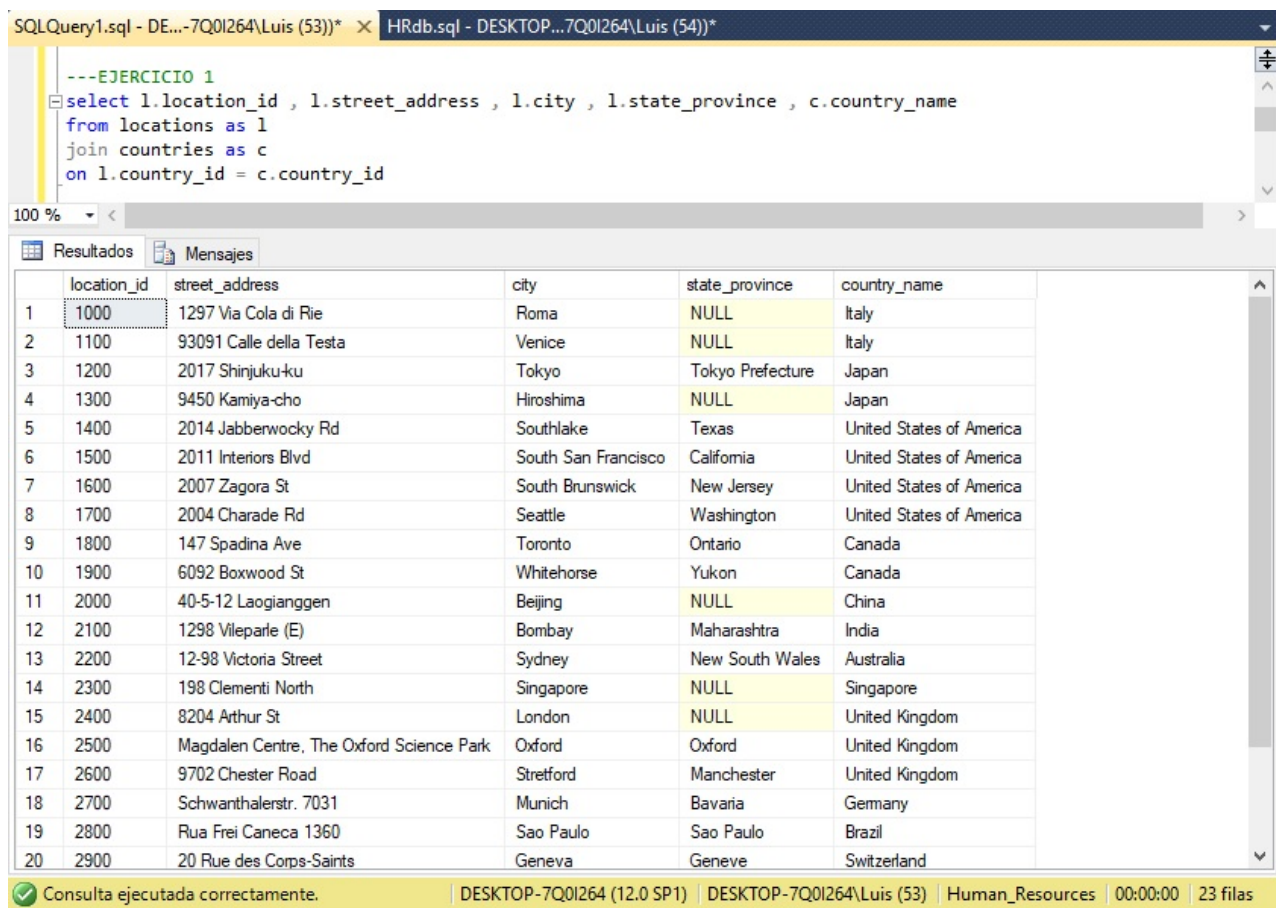
```
select salman.minimo, salman.manager_id from(select min(salary)as 'Minimo',manager_id from employees where salary>6000 group by manager_id) as salman order by salman.minimo;
```

|    | minimo  | manager_id |
|----|---------|------------|
| 1  | 6100.00 | 148        |
| 2  | 6200.00 | 149        |
| 3  | 6200.00 | 147        |
| 4  | 6500.00 | 100        |
| 5  | 6500.00 | 101        |
| 6  | 6900.00 | 108        |
| 7  | 7000.00 | 145        |
| 8  | 7000.00 | 146        |
| 9  | 8300.00 | 205        |
| 10 | 9000.00 | 102        |
| 11 | 24000   | NULL       |

## 8. Actividad No 08 – Enlaces

1. El departamento de Recursos Humanos requiere un reporte que muestre las direcciones de todos los departamentos. Utilizar las tablas LOCATIONS y COUNTRIES. Mostrar el ID de la Ubicación (location\_id), dirección (street\_address), ciudad (city), estado o provincia (state\_province) y país (country\_name).

```
select l.location_id , l.street_address , l.city , l.state_province , c.country_name
from locations as l
join countries as c
on l.country_id = c.country_id
```



The screenshot shows a SQL Developer window with a query titled '---EJERCICIO 1'. The query is a join between the LOCATIONS and COUNTRIES tables. The results are displayed in a table with 20 rows and 5 columns: location\_id, street\_address, city, state\_province, and country\_name. The status bar at the bottom indicates the query was executed successfully.

|    | location_id | street_address                           | city                | state_province   | country_name             |
|----|-------------|--|---------------------|------------------|--------------------------|
| 1  | 1000        | 1297 Via Cola di Rie                     | Roma                | NULL             | Italy                    |
| 2  | 1100        | 93091 Calle della Testa                  | Venice              | NULL             | Italy                    |
| 3  | 1200        | 2017 Shinjuku-ku                         | Tokyo               | Tokyo Prefecture | Japan                    |
| 4  | 1300        | 9450 Kamiya-cho                          | Hiroshima           | NULL             | Japan                    |
| 5  | 1400        | 2014 Jabberwocky Rd                      | Southlake           | Texas            | United States of America |
| 6  | 1500        | 2011 Interiors Blvd                      | South San Francisco | California       | United States of America |
| 7  | 1600        | 2007 Zagora St                           | South Brunswick     | New Jersey       | United States of America |
| 8  | 1700        | 2004 Charade Rd                          | Seattle             | Washington       | United States of America |
| 9  | 1800        | 147 Spadina Ave                          | Toronto             | Ontario          | Canada                   |
| 10 | 1900        | 6092 Boxwood St                          | Whitehorse          | Yukon            | Canada                   |
| 11 | 2000        | 40-5-12 Laogianggen                      | Beijing             | NULL             | China                    |
| 12 | 2100        | 1298 Vileparle (E)                       | Bombay              | Maharashtra      | India                    |
| 13 | 2200        | 12-98 Victoria Street                    | Sydney              | New South Wales  | Australia                |
| 14 | 2300        | 198 Clementi North                       | Singapore           | NULL             | Singapore                |
| 15 | 2400        | 8204 Arthur St                           | London              | NULL             | United Kingdom           |
| 16 | 2500        | Magdalen Centre, The Oxford Science Park | Oxford              | Oxford           | United Kingdom           |
| 17 | 2600        | 9702 Chester Road                        | Stretford           | Manchester       | United Kingdom           |
| 18 | 2700        | Schwanthalerstr. 7031                    | Munich              | Bavaria          | Germany                  |
| 19 | 2800        | Rua Frei Caneca 1360                     | Sao Paulo           | Sao Paulo        | Brazil                   |
| 20 | 2900        | 20 Rue des Corps-Saints                  | Geneva              | Geneve           | Switzerland              |

Consulta ejecutada correctamente. | DESKTOP-7Q0I264 (12.0 SP1) | DESKTOP-7Q0I264\Luis (53) | Human\_Resources | 00:00:00 | 23 filas

2. El departamento de Recursos Humanos necesita un reporte de todos empleados, que muestres los apellidos de empleado (last\_name), el No de departamento (department\_id) y el nombre del departamento (department\_name) al cual pertenece.

```
select e.last_name , d.department_id , d.department_name from employees as e
left join departments as d
on e.department_id = d.department_id order by d.department_name;
```



SQLQuery1.sql - DE...-7Q0I264\Luis (53))\* x HRdb.sql - DESKTOP...7Q0I264\Luis (54))\*

```

---EJERCICIO 2
select e.last_name , d.department_id , d.department_name from employees as e
left join departments as d
on e.department_id = d.department_id order by d.department_name;

```

100 %

Resultados Mensajes

|    | last_name | department_id | department_name |
|----|-----------|---------------|-----------------|
| 1  | Grant     | NULL          | NULL            |
| 2  | Higgins   | 110           | Accounting      |
| 3  | Gietz     | 110           | Accounting      |
| 4  | Whalen    | 10            | Administration  |
| 5  | King      | 90            | Executive       |
| 6  | Kochhar   | 90            | Executive       |
| 7  | De Haan   | 90            | Executive       |
| 8  | Greenberg | 100           | Finance         |
| 9  | Faviet    | 100           | Finance         |
| 10 | Chen      | 100           | Finance         |
| 11 | Sciarra   | 100           | Finance         |
| 12 | Uman      | 100           | Finance         |
| 13 | Popp      | 100           | Finance         |
| 14 | Mavris    | 40            | Human Resources |
| 15 | Hunold    | 60            | IT              |
| 16 | Ernst     | 60            | IT              |
| 17 | Austin    | 60            | IT              |
| 18 | Pataballa | 60            | IT              |
| 19 | Lorentz   | 60            | IT              |
| 20 | Hartstein | 20            | Marketing       |
| 21 | ...       | 20            | Marketing       |

Consulta ejecutada correctamente. DESKTOP-7Q0I264 (12.0 SP1) DESKTOP-7Q0I264\Luis (53) Human\_Resources 00:00:00 107 filas

3. El departamento de Recursos Humanos necesita un reporte de los empleados de la ciudad de Toronto. Mostrar los Apellidos, Puesto, No de Departamento y Nombre de Departamento de todos los empleados que trabajan en Toronto.

```

select e.last_name , e.department_id, j.job_title, d.department_name , l.city
from employees as e
left join jobs as j
on e.job_id = j.job_id
join departments as d
on e.department_id=d.department_id
join locations as l
on d.location_id = l.location_id
where l.city='Toronto';

```

SQLQuery1.sql - DE...-7Q0I264\Luis (53))\* X HRdb.sql - DESKTOP...7Q0I264\Luis (54))\*

```

----EJERCICIO 3|
select e.last_name , e.department_id, j.job_title, d.department_name , l.city
from employees as e
left join jobs as j
on e.job_id = j.job_id
join departments as d
on e.department_id=d.department_id
join locations as l
on d.location_id = l.location_id
where l.city='Toronto';

```

100 %

Resultados Mensajes

|   | last_name | department_id | job_title                | department_name | city    |
|---|-----------|---------------|--------------------------|-----------------|---------|
| 1 | Hartstein | 20            | Marketing Manager        | Marketing       | Toronto |
| 2 | Fay       | 20            | Marketing Representative | Marketing       | Toronto |

4. Crear un reporte que muestre los Apellidos y No de Identificación de los empleados, asimismo también debe mostrarse el Apellido y No de Identificación de su Administrador.

```

SELECT e.employee_id 'ID_Empleado', e.last_name 'Empleado',
m.employee_id 'ID_Manager', m.last_name 'Manager'
FROM employees e
join employees m
ON (e.manager_id = m.employee_id)

```

sql ACTIVIDAD 8.sql...-7Q0I264\Luis (51))\* X

```

----EJERCICIO 4
SELECT e.employee_id 'ID_Empleado', e.last_name 'Empleado',
m.employee_id 'ID_Manager', m.last_name 'Manager'
FROM employees e
join employees m
ON (e.manager_id = m.employee_id)

```

100 %

Resultados Mensajes

|    | ID_Empleado | Empleado  | ID_Manager | Manager |
|----|-------------|-----------|------------|---------|
| 1  | 101         | Kochhar   | 100        | King    |
| 2  | 102         | De Haan   | 100        | King    |
| 3  | 114         | Raphaely  | 100        | King    |
| 4  | 120         | Weiss     | 100        | King    |
| 5  | 121         | Fripp     | 100        | King    |
| 6  | 122         | Kaufling  | 100        | King    |
| 7  | 123         | Vollman   | 100        | King    |
| 8  | 124         | Mourgos   | 100        | King    |
| 9  | 145         | Russell   | 100        | King    |
| 10 | 146         | Partners  | 100        | King    |
| 11 | 147         | Erazuriz  | 100        | King    |
| 12 | 148         | Cambrault | 100        | King    |
| 13 | 149         | Zlotkey   | 100        | King    |
| 14 | 201         | Hartstein | 100        | King    |
| 15 | 200         | Whalen    | 101        | Kochhar |
| 16 | 203         | Mavris    | 101        | Kochhar |
| 17 | 204         | Baer      | 101        | Kochhar |
| 18 | 205         | Higgins   | 101        | Kochhar |
| 19 | 108         | Greenh    | 101        | Kochhar |

Consulta ejecutada correctamente. DESKTOP-7Q0I264 (12.0 SP1) DESKTOP-7Q0I264\Luis (51) Human\_Resources 00:00:00 106 filas

- Modificar la consulta anterior para que incluya también a los empleados quienes no tienen Administrador asignado.

```

SELECT e.employee_id 'ID_Empleado', e.last_name 'Empleado',
m.employee_id 'ID_Manager', m.last_name 'Manager'
FROM employees e
left outer join employees m
ON (e.manager_id = m.employee_id)

```

sql ACTIVIDAD 8.sql...-7Q0I264\Luis (51))\*

```

----EJERCICIO 5
SELECT e.employee_id 'ID_Empleado', e.last_name 'Empleado',
m.employee_id 'ID_Manager', m.last_name 'Manager'
FROM employees e
left outer join employees m
ON (e.manager_id = m.employee_id)

```

100 %

Resultados Mensajes

|    | ID_Empleado | Empleado  | ID_Manager | Manager |
|----|-------------|-----------|------------|---------|
| 1  | 100         | King      | NULL       | NULL    |
| 2  | 101         | Kochhar   | 100        | King    |
| 3  | 102         | De Haan   | 100        | King    |
| 4  | 114         | Raphaely  | 100        | King    |
| 5  | 120         | Weiss     | 100        | King    |
| 6  | 121         | Fripp     | 100        | King    |
| 7  | 122         | Kaufling  | 100        | King    |
| 8  | 123         | Vollman   | 100        | King    |
| 9  | 124         | Mourgos   | 100        | King    |
| 10 | 145         | Russell   | 100        | King    |
| 11 | 146         | Partners  | 100        | King    |
| 12 | 147         | Errazuriz | 100        | King    |
| 13 | 148         | Cambrault | 100        | King    |
| 14 | 149         | Zlotkey   | 100        | King    |
| 15 | 201         | Hartstein | 100        | King    |
| 16 | 200         | Whalen    | 101        | Kochhar |
| 17 | 203         | Mavris    | 101        | Kochhar |
| 18 | 204         | Baer      | 101        | Kochhar |
| 19 | 205         | Higgins   | 101        | Kochhar |
| 20 | 100         | King      | NULL       | NULL    |

Consulta ejecutada correctamente. DESKTOP-7Q0I264 (12.0 SP1) DESKTOP-7Q0I264\Luis (51) Human\_Resources 00:00:00 107 filas

6. Crear un reporte que muestre los No de Departamento y Apellidos de todos los empleados, asimismo adicionar una columna con los Apellidos de todos empleados que trabajan en el mismo departamento. Etiquetar esta columna como Colega.

```

select e.department_id 'DEPARTAMENTO', e.last_name 'EMPLEADO',
d.last_name 'COLEGA'
from employees e
join employees d
on (e.department_id=d.department_id) and e.last_name!=d.last_name;

```

sql ACTIVIDAD 8.sql...-7Q0I264\Luis (51))\*

```

----EJERCICIO 6
select e.department_id 'DEPARTMENT', e.last_name 'EMPLEADO',
d.last_name 'COLEGA'
from employees e
join employees d
on (e.department_id=d.department_id) and e.last_name!=d.last_name;

```

100 %

Resultados Mensajes

|    | DEPARTMENT | EMPLEADO | COLEGA    |
|----|------------|----------|-----------|
| 61 | 30         | Raphaely | Colmen... |
| 62 | 30         | Khoo     | Raphaely  |
| 63 | 30         | Khoo     | Baida     |
| 64 | 30         | Khoo     | Tobias    |
| 65 | 30         | Khoo     | Himuro    |
| 66 | 30         | Khoo     | Colmen... |
| 67 | 30         | Baida    | Raphaely  |
| 68 | 30         | Baida    | Khoo      |
| 69 | 30         | Baida    | Tobias    |
| 70 | 30         | Baida    | Himuro    |
| 71 | 30         | Baida    | Colmen... |
| 72 | 30         | Tobias   | Raphaely  |
| 73 | 30         | Tobias   | Khoo      |
| 74 | 30         | Tobias   | Baida     |
| 75 | 30         | Tobias   | Himuro    |
| 76 | 30         | Tobias   | Colmen... |
| 77 | 30         | Himuro   | Raphaely  |
| 78 | 30         | Himuro   | Khoo      |

Consulta ejecutada correctamente. DESKTOP-7Q0I264 (12.0 SP1) DESKTOP-7Q0I264\Luis (51) Human\_Resources 00:00:00 3188 filas

7. El departamento de Recursos Humanos requiere un reporte de todo el personal que fue contratado después del empleado apellidado 'Davies'. Crear un reporte que muestre el apellidos y fecha de contratación de todo los empleados contratado después de 'Davies'.

```

SELECT e.first_name, e.last_name, e.hire_date
FROM employees e
JOIN employees davies
ON (davies.last_name = 'Davies')
WHERE davies.hire_date < e.hire_date;

```

Scripts - Consulta...Lenovo-PC\win (54)) \* Script - Base de D...Lenovo-PC\win (56))

/\*ACTIVIDAD 8 - ENLACES

EJERCICIO 7

El departamento de Recursos Humanos requiere un reporte de todo el personal que fue contratado después del empleado apellidado 'Davies'. Crear un reporte que muestre el apellidos y fecha de contratación de todo los empleados contratado después de 'Davies'.\*/

```

SELECT e.first_name, e.last_name, e.hire_date
FROM employees e
JOIN employees davies
ON (davies.last_name = 'Davies')
WHERE davies.hire_date < e.hire_date;

```

100 %

Results Messages

|    | first_name  | last_name | hire_date  |
|----|-------------|-----------|------------|
| 1  | Neena       | Kochhar   | 2005-09-21 |
| 2  | Alexander   | Hunold    | 2006-01-03 |
| 3  | Bruce       | Ernst     | 2007-05-21 |
| 4  | David       | Austin    | 2005-06-25 |
| 5  | Valli       | Pataballa | 2006-02-05 |
| 6  | Diana       | Lorentz   | 2007-02-07 |
| 7  | John        | Chen      | 2005-09-28 |
| 8  | Ismael      | Sciarra   | 2005-09-30 |
| 9  | Jose Manuel | Uman      | 2006-03-07 |
| 10 | Luis        | Popp      | 2007-12-07 |
| 11 | Shelli      | Baida     | 2005-12-24 |
| 12 | Sigal       | Tobias    | 2005-07-24 |
| 13 | Guy         | Himuro    | 2006-11-15 |

Query executed successfully. | Lenovo-PC (13.0 RTM) | Lenovo-PC\win (54) | Human\_Resources | 00:00:18 | 81 rows

8. El departamento de Recursos Humanos requiere de un reporte que el apellido del empleado, fecha de contratación del empleado, apellido del administrador, fecha de contratación del administrador. Para todos aquellos empleados que fueron contratados antes que sus Administradores.

Ingresa el siguiente código:

```

SELECT e.last_name 'EMPLEADO', e.hire_date 'FECHA CONTRATACION' , j.last_name
'ADMINISTRADOR',
j.hire_date 'FECHA CONTRATACION ADMINISTRADOR'
from employees e
join employees j
on e.manager_id=j.employee_id
and e.hire_date < j.hire_date
order by e.hire_date;

```

Scripts - Consulta...Lenovo-PC\win (54) X Script - Base de D...Lenovo-PC\win (56)

```

/*
EJERCICIO 8

El departamento de Recursos Humanos requiere de un reporte que el apellido del empleado, fecha de
contratación del empleado, apellido del administrador, fecha de contratación del administrador. Para
todos aquellos empleados que fueron contratados antes que sus Administradores.*/

SELECT e.last_name 'EMPLEADO', e.hire_date 'FECHA CONTRATACION', j.last_name 'ADMINISTRADOR',
j.hire_date 'FECHA CONTRATACION ADMINISTRADOR'
FROM employees e
JOIN employees j
ON e.manager_id=j.employee_id
AND e.hire_date < j.hire_date
ORDER BY e.hire_date;

```

100 %

Results Messages

|    | EMPLEADO  | FECHA CONTRATACION | ADMINISTRADOR | FECHA CONTRATACION ADMINISTRADOR |
|----|-----------|--------------------|---------------|----------------------------------|
| 1  | De Haan   | 2001-01-13         | King          | 2003-06-17                       |
| 2  | Mavris    | 2002-06-07         | Kochhar       | 2005-09-21                       |
| 3  | Baer      | 2002-06-07         | Kochhar       | 2005-09-21                       |
| 4  | Higgins   | 2002-06-07         | Kochhar       | 2005-09-21                       |
| 5  | Faviet    | 2002-08-16         | Greenberg     | 2002-08-17                       |
| 6  | Greenberg | 2002-08-17         | Kochhar       | 2005-09-21                       |
| 7  | Raphaely  | 2002-12-07         | King          | 2003-06-17                       |
| 8  | Kaufling  | 2003-05-01         | King          | 2003-06-17                       |
| 9  | Ladwig    | 2003-07-14         | Vollman       | 2005-10-10                       |
| 10 | Whalen    | 2003-09-17         | Kochhar       | 2005-09-21                       |
| 11 | Rajs      | 2003-10-17         | Mourgos       | 2007-11-16                       |
| 12 | Sarchand  | 2004-01-27         | Frippe        | 2005-04-10                       |
| 13 | King      | 2004-01-30         | Partners      | 2005-01-05                       |

Query executed successfully. LENOVO-PC (13.0 RTM) Lenovo-PC\win (54) Human\_Resources 00:00:01 37 rows

## 9. Actividad No 09 – SubConsultas

1. El departamento de Recursos Humanos requiere una consulta que pregunte al usuario por el Apellido del empleado, Luego la consulta deberá mostrar los Apellidos y Fecha de Contratación de todos los empleados del mismo departamento excluyendo o con excepción del empleado el cual ha sido proporcionado su apellido reporte que muestre las direcciones de todos los departamentos.

– para el desarrollo se cambio apellido por id de empleado

```
DECLARE @depid INT;  
DECLARE @empid INT;
```

– se procede a leer el id de empleado

```
SET @empid = 110
```

–obteniendo id de departamento de empleado

```
SET @depid = (SELECT emp.department_id  
FROM employees as emp  
WHERE emp.employee_id=@empid);
```

–todos los empleados del mismo departamento excluyendo al empleado ingresado anteriormente

```
SELECT emp.employee_id,  
emp.last_name,  
emp.hire_date,  
emp.department_id  
FROM employees AS emp  
WHERE emp.department_id = @depid  
AND emp.employee_id !=@empid;
```



The screenshot shows a SQL Server Enterprise Manager window with two panes. The top pane is a script window titled 'Script - Base de D...Lenovo-PC(win (56))' containing a T-SQL query. The query is a stored procedure that takes an employee ID as input and returns the last names and hire dates of all other employees in the same department. The bottom pane is the 'Results' window, which shows the output of the query as a table with 5 rows and 4 columns: employee\_id, last\_name, hire\_date, and department\_id.

```

--ACTIVIDAD 9 - SUBCONSULTAS - EJERCICIO 1

El departamento de Recursos Humanos requiere una consulta que pregunte al usuario por el Apellido
del empleado. Luego la consulta deberá mostrar los Apellidos y Fecha de Contratación de todos los
empleados del mismo departamento excluyendo o con excepción del empleado el cual ha sido
proporcionado su apellido reporte que muestre las direcciones de todos los departamentos.
*/

--para el desarrollo se cambio apellido por id de empleado

DECLARE @depid INT;
DECLARE @empid INT;

--leyendo id de empleado
SET @empid = 110

--obteniendo id de departamento de empleado
SET @depid = (SELECT emp.department_id
FROM employees as emp
WHERE emp.employee_id=@empid);

--todos los empleados del mismo departamento excluyendo al empleado ingresado anteriormente
SELECT emp.employee_id,
emp.last_name,
emp.hire_date,
emp.department_id
FROM employees AS emp WHERE emp.department_id = @depid AND emp.employee_id != @empid;

```

| employee_id | last_name | hire_date  | department_id |
|-------------|-----------|------------|---------------|
| 108         | Greenberg | 2002-08-17 | 100           |
| 109         | Faviet    | 2002-08-16 | 100           |
| 111         | Sciarra   | 2005-09-30 | 100           |
| 112         | Uman      | 2006-03-07 | 100           |
| 113         | Popp      | 2007-12-07 | 100           |

Query executed successfully. | LENOVO-PC (13.0 RTM) | Lenovo-PC(win (54)) | Human\_Resources | 00:00:00 | 5 rows

2. Crear un reporte que muestre el No del Empleado, Apellidos y Salarios de todos los empleados que tienen un salario superior al promedio de salarios de todos los empleados. Ordenar los resultados por el Salario de forma ascendente.

–Se considera 'N de empleado' como 'id de empleado'

–Obteniendo promedio de salario

DECLARE @prom DECIMAL (8,2); – Variable de promedio

SET @prom = (SELECT AVG(salary) FROM employees);

–Todos los empleados con sueldo superior al promedio

SELECT emp.employee\_id,

emp.last\_name,

emp.salary

FROM employees AS emp

WHERE emp.salary > @prom;

The screenshot shows a SQL Developer window with a script titled 'EJERCICIO 2'. The script's purpose is to create a report of employee IDs, last names, and salaries for those earning above the average salary, ordered by salary in ascending order. The script includes comments in Spanish explaining the steps: considering employee ID as the identifier, calculating the average salary, and then selecting employees whose salary is greater than this average.

```

/*
EJERCICIO 2

Crear un reporte que muestre el N° del Empleado, Apellidos y Salarios de todos los empleados que
tienen un salario superior al promedio de salarios de todos los empleados. Ordenar los resultados por
el Salario de forma ascendente.*/

--Se considera "N° de empleado" como "id de empleado"
--Obteniendo promedio de salario

DECLARE @prom DECIMAL (8,2); -- Variable de promedio
SET @prom = (SELECT AVG(salary) FROM employees);

--Todos los empleados con sueldo superior al promedio
SELECT emp.employee_id,
       emp.last_name,
       emp.salary
FROM employees AS emp
WHERE emp.salary > @prom;

```

The results pane shows the following data:

| employee_id | last_name | salary   |
|-------------|-----------|----------|
| 100         | King      | 24000.00 |
| 101         | Kochhar   | 17000.00 |
| 102         | De Haan   | 17000.00 |
| 103         | Hunold    | 9000.00  |
| 108         | Greenberg | 12008.00 |
| 109         | Faviet    | 9000.00  |
| 110         | Chen      | 8200.00  |
| 111         | Sciarra   | 7700.00  |

The status bar at the bottom indicates the query was executed successfully on the 'Human\_Resources' schema, returning 51 rows.

- Realizar un reporte que muestre el No de Empleado y Apellidos de todos los empleados quienes trabajan en el departamento de cualquier empleado que su apellido contenga la letra 'u'.

–Se considera 'N de empleado' como 'id de empleado'

–Obtener los id de departamentos de los empleados que contengan 'u' en su apellido

```

SELECT DISTINCT department_id
FROM employees
WHERE last_name LIKE '%u%'

```

–Obtener todos los empleados que laboren en alguno de los departamentos hallados anteriormente

```

SELECT emp.employee_id,
       emp.last_name,
       emp.department_id
FROM employees AS emp
JOIN (SELECT DISTINCT department_id
      FROM employees
      WHERE last_name LIKE '%u%' ) AS depid
ON emp.department_id=depid.department_id

```

Scripts - Consulta...Lenovo-PC\win (54)\* Script - Base de D...Lenovo-PC\win (56)

EJERCICIO 3

Realizar un reporte que muestre el N° de Empleado y Apellidos de todos los empleados quienes trabajan en el departamento de cualquier empleado que su apellido comience con "u".

--Se considera "N° de empleado" como "id de empleado"

--Obtener los id de departamentos de los empleados que contengan "u" en su apellido

```

SELECT DISTINCT department_id
FROM employees
WHERE last_name LIKE '%u%'

```

--Obtener todos los empleados que laboren en alguno de los departamentos hallados anteriormente

```

SELECT emp.employee_id,
       emp.last_name,
       emp.department_id
FROM employees AS emp
JOIN (SELECT DISTINCT department_id
      FROM employees
      WHERE last_name LIKE '%u%') AS depid
ON emp.department_id=depid.department_id

```

100 %

Results Messages

|   | department_id |
|---|---------------|
| 1 | 30            |
| 2 | 50            |
| 3 | 60            |
| 4 | 80            |

|   | employee_id | last_name | department_id |
|---|-------------|-----------|---------------|
| 1 | 114         | Raphaely  | 30            |
| 2 | 115         | Khoo      | 30            |
| 3 | 116         | Baida     | 30            |
| 4 | 117         | Tobias    | 30            |
| 5 | 118         | Himuro    | 30            |

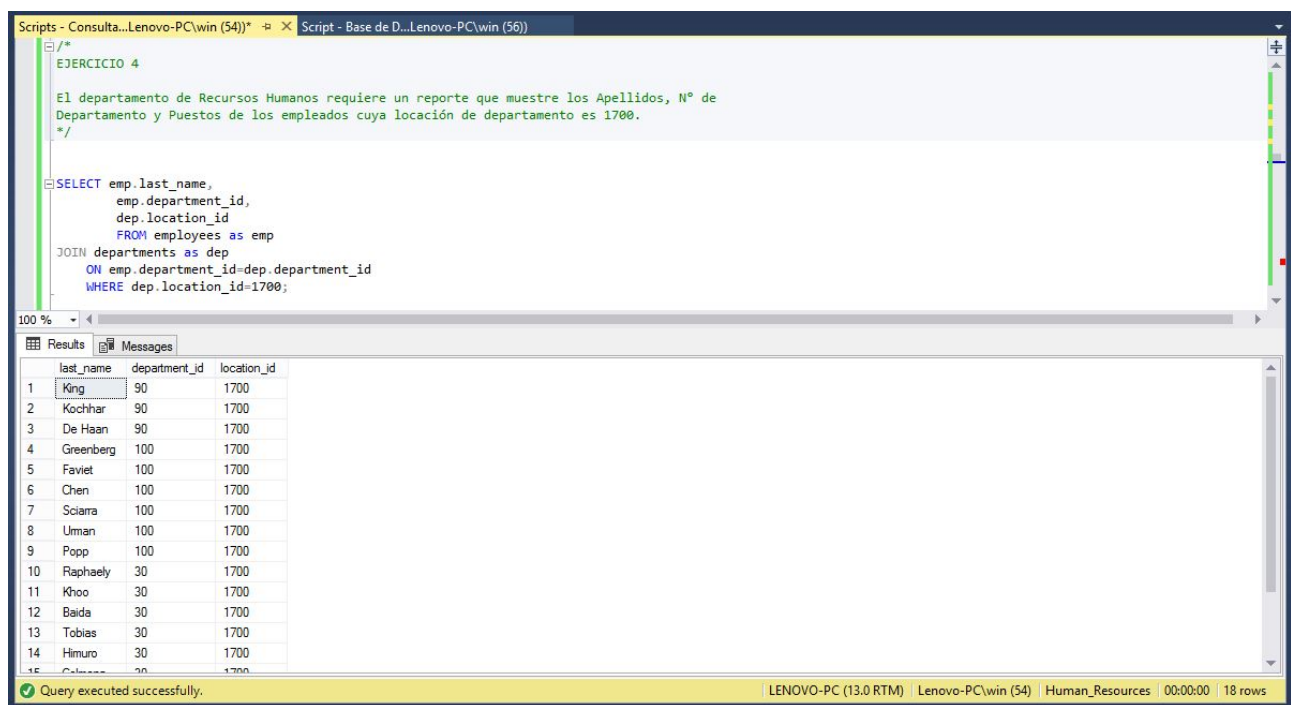
Query executed successfully. | LENOVO-PC (13.0 RTM) | Lenovo-PC\win (54) | Human\_Resources | 00:00:00 | 101 rows

4. El departamento de Recursos Humanos requiere un reporte que muestre los Apellidos, No de Departamento y Puestos de los empleados cuya locación de departamento es 1700.

```

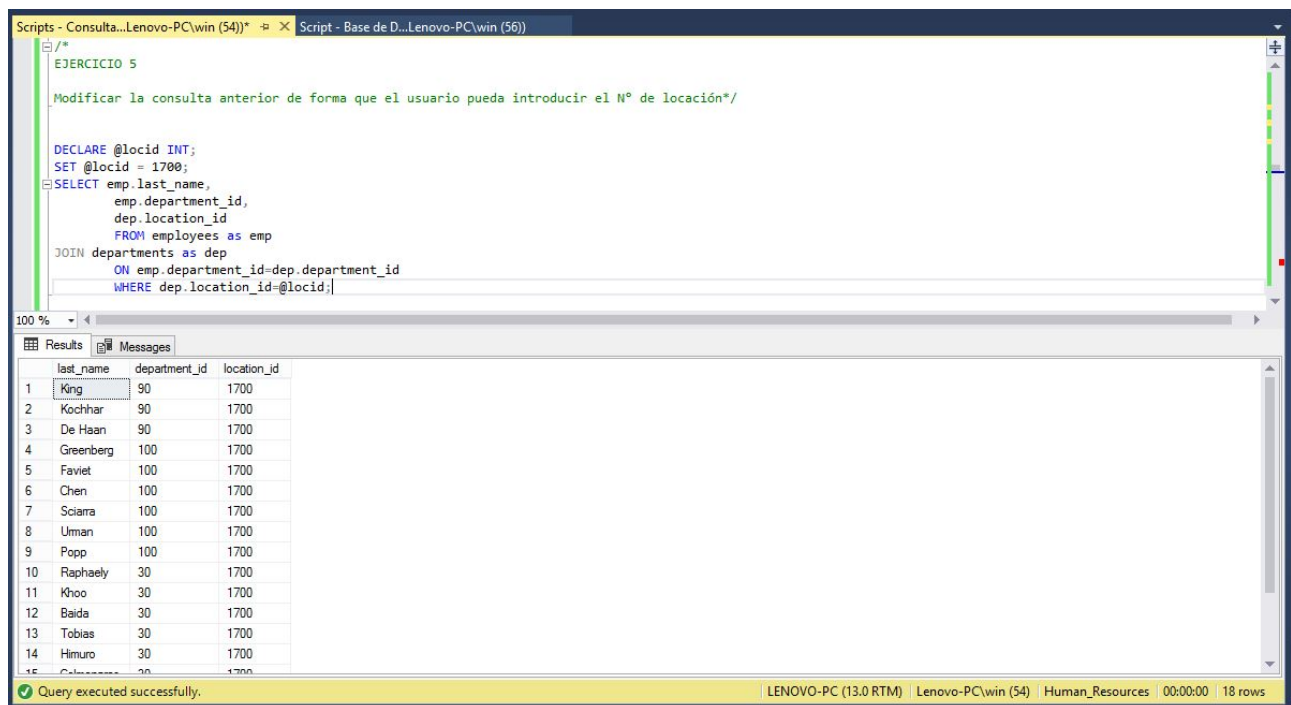
SELECT emp.last_name,
       emp.department_id,
       dep.location_id
FROM employees as emp
JOIN departments as dep
ON emp.department_id=dep.department_id
WHERE dep.location_id=1700;

```



5. Modificar la consulta anterior de forma que el usuario pueda introducir el No de locación.

```
DECLARE @locid INT;  
SET @locid = 1700;  
SELECT emp.last_name,  
       emp.department_id,  
       dep.location_id  
FROM employees as emp  
JOIN departments as dep  
  ON emp.department_id=dep.department_id  
 WHERE dep.location_id=@locid;
```



6. Crear un reporte para el departamento de Recursos Humanos que muestre los Apellidos y Salarios de todos los empleados cuyo Administrador apellide 'King'.

–conseguir id de empleado que lleven como apellido KING

```

SELECT employee_id,
last_name
FROM employees
WHERE last_name='KING';

```

–conseguir id de departamentos que coincidan con el manager\_id con employee\_id

```

SELECT dep.department_id
FROM departments AS dep
JOIN (SELECT employee_id,
last_name
FROM employees
WHERE last_name='KING') as manking
ON dep.manager_id=manking.employee_id

```

–Obtener los apellidos y salarios de empleados que tengan como id de departamneto el/los id de departamento hallados

```

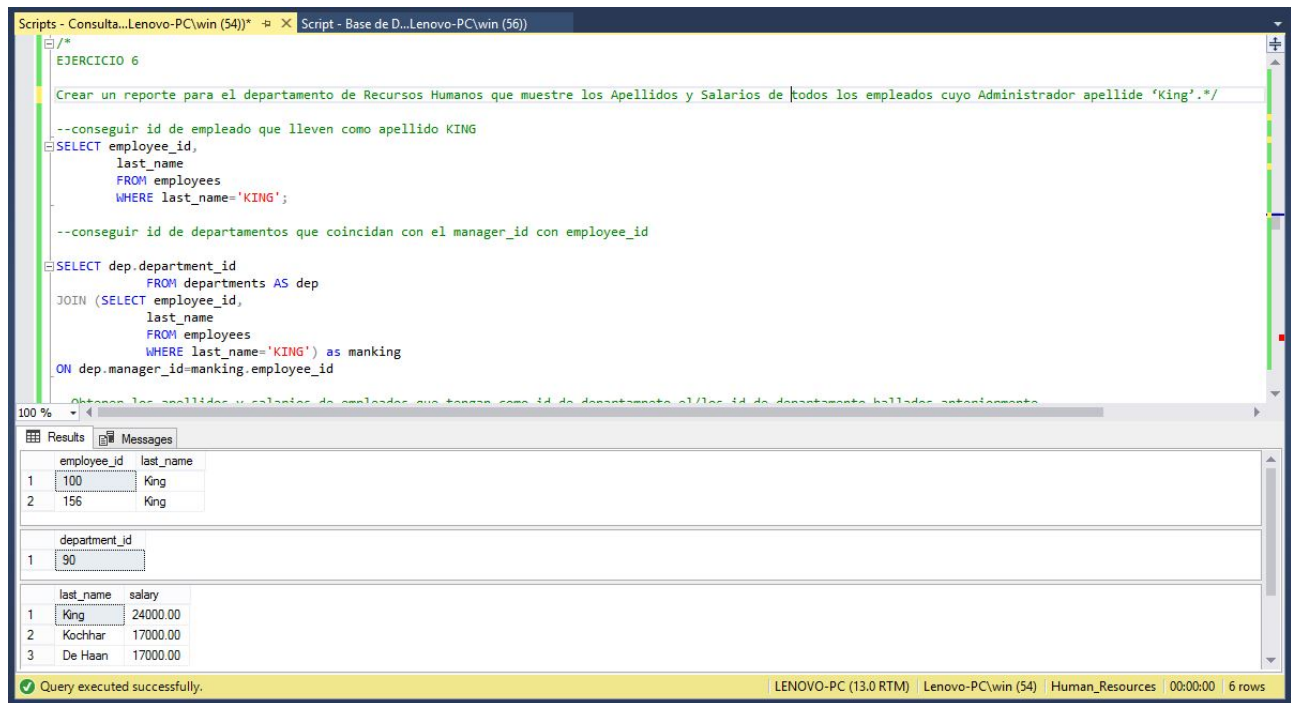
SELECT emp.last_name,
emp.salary
FROM employees AS emp
JOIN (SELECT dep.department_id
FROM departments AS dep
JOIN (SELECT employee_id,

```

```

last_name
FROM employees
WHERE last_name='KING') AS manking
ON dep.manager_id = manking.employee_id) AS depking
ON emp.department_id=depking.department_id;

```



7. Crear un reporte para el departamento de Recursos Humanos que muestre el No de Departamento, Apellidos, Puestos de todos los empleados en el departamento 'Executive'.

```

SELECT * from employees where department_id=90;
SELECT * from jobs;
SELECT * from departments where department_name='executive';

```

–consiguiendo empleados con nombre de puesto

```

SELECT emp.department_id,
emp.last_name,
jobs.job_title
FROM employees AS emp
JOIN jobs
ON emp.job_id=jobs.job_id;

```

–Conseguir a los empleado con departamento categoria Executive

```

SELECT empnomjob.department_id,
empnomjob.last_name,
empnomjob.job_title
FROM departments

```

```

JOIN (SELECT emp.department_id,
emp.last_name,
jobs.job_title
FROM employees AS emp
JOIN jobs
ON emp.job_id=jobs.job_id) AS empnomjob
ON empnomjob.department_id=departments.department_id
WHERE department_name='executive';

```

Scripts - Consulta...Lenovo-PC\win (54) - Script - Base de D...Lenovo-PC\win (56)

```

/*
EJERCICIO 7

Crear un reporte para el departamento de Recursos Humanos que muestre el N° de Departamento, Apellidos, Puestos de todos los empleados en el departamento 'Execut:

SELECT * from employees where department_id=90;
SELECT * from jobs;
SELECT * from departments where department_name='executive';

--consiguiendo empleados con nombre de puesto
SELECT emp.department_id,
emp.last_name,
jobs.job_title
FROM employees AS emp
JOIN jobs
ON emp.job_id=jobs.job_id;

--Conseguir a los empleado con departamento Executive
SELECT empnomjob.department_id,
empnomjob.last_name,
empnomjob.job_title
FROM empnomjob
JOIN departments
ON empnomjob.department_id=departments.department_id;

```

100 %

Results Messages

| employee_id | first_name | last_name | email   | phone_number | hire_date    | job_id     | salary  | commission_pct | manager_id | department_id |    |
|-------------|------------|-----------|---------|--------------|--------------|------------|---------|----------------|------------|---------------|----|
| 1           | 100        | Steven    | King    | SKING        | 515.123.4567 | 2003-06-17 | AD_PRES | 24000.00       | NULL       | NULL          | 90 |
| 2           | 101        | Neena     | Kochhar | NKOCHHAR     | 515.123.4568 | 2005-09-21 | AD_VP   | 17000.00       | NULL       | 100           | 90 |
| 3           | 102        | Lex       | De Haan | LDEHAAN      | 515.123.4569 | 2001-01-13 | AD_VP   | 17000.00       | NULL       | 100           | 90 |

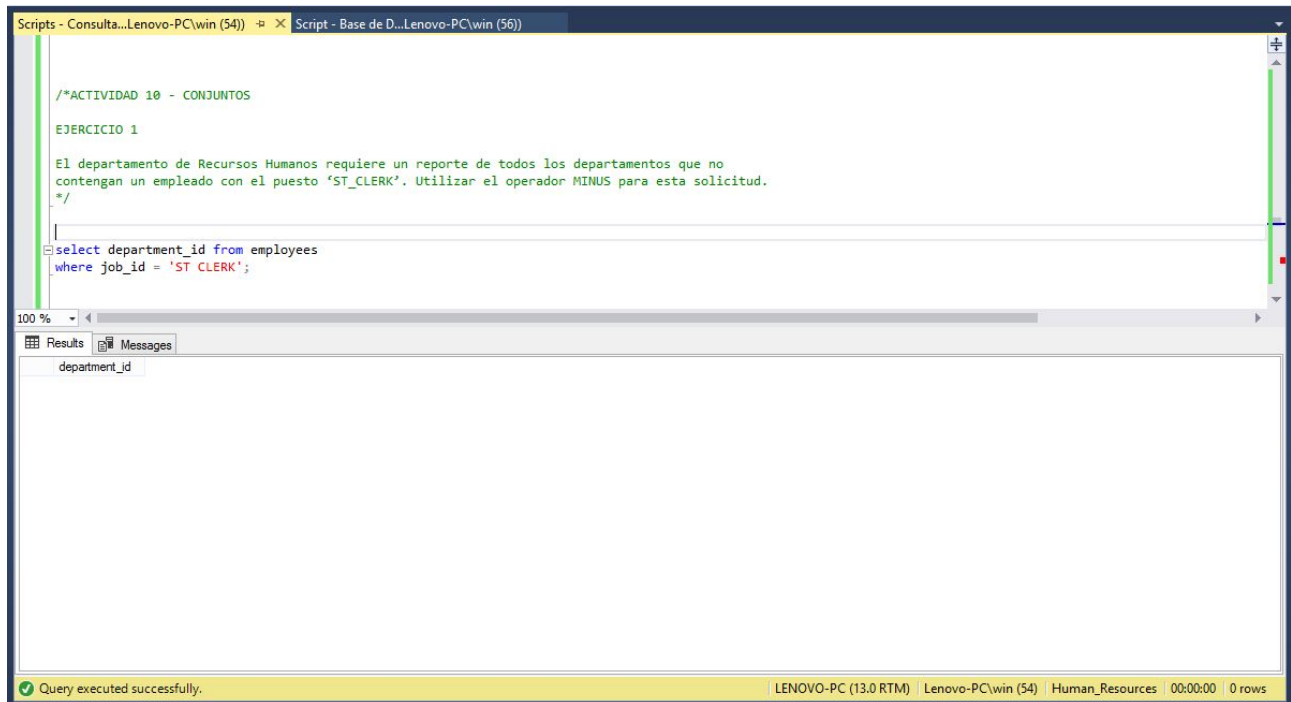
| job_id | job_title  | min_salary | max_salary |
|--------|------------|------------|------------|
| 1      | AC_ACCOUNT | 4200       | 9000       |
| 2      | AC_MGR     | 8200       | 16000      |
| 3      | AD_ASST    | 3000       | 6000       |
| 4      | AD_PRES    | 20080      | 40000      |
| 5      | AD_VP      | 15000      | 30000      |

Query executed successfully. | LENOVO-PC (13.0 RTM) | Lenovo-PC\win (54) | Human\_Resources | 00:00:01 | 133 rows

## 10. Actividad No 10 – Conjuntos

1. El departamento de Recursos Humanos requiere un reporte de todos los departamentos que no contengan un empleado con el puesto 'ST\_CLERK'. Utilizar el operador MINUS o EXCEPT para esta solicitud.

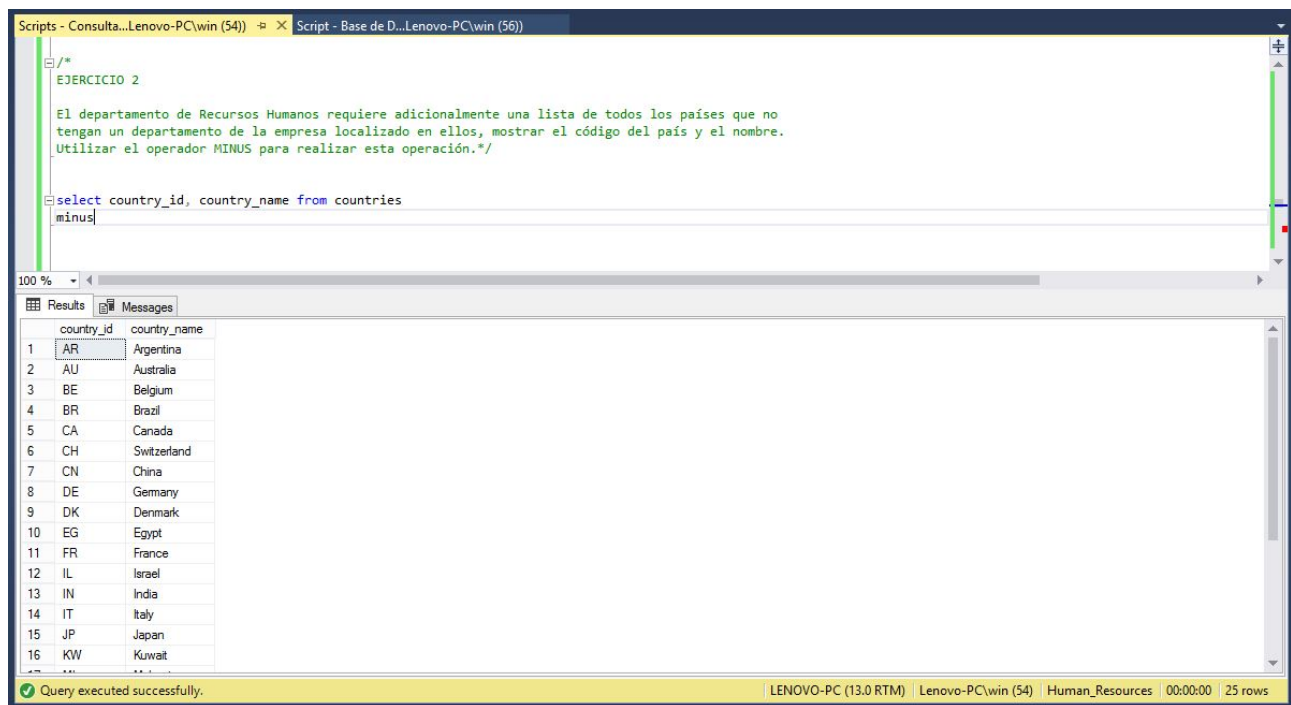
```
select department_id from employees
where job_id = 'ST CLERK';
```



2. El departamento de Recursos Humanos requiere adicionalmente una lista de todos los países que no tengan un departamento de la empresa localizado en ellos, mostrar el código del país y el nombre. Utilizar el operador MINUS o EXCEPT para realizar esta operación.

```
select country_id, country_name from countries
minus
```





3. Se necesita una lista de puestos de los departamentos 10, 50 y 20, en ese orden, mostrar el código del puesto y código del departamento. Utilizar el operador UNION ALL.

```
select distinct job_id, department_id from employees
where (department_id=10)
union
select distinct job_id, department_id from employees
where (department_id=50)
union
select distinct job_id, department_id from employees
where (department_id=20);
```

Scripts - Consulta...Lenovo-PC\win (54) X Script - Base de D...Lenovo-PC\win (56)

```

/*
EJERCICIO 3

Se necesita una lista de puestos de los departamentos 10, 50 y 20, en ese orden, mostrar el código del
puesto y código del departamento. Utilizar el operador UNION ALL*/

select distinct job_id, department_id from employees
where (department_id=10)
union
select distinct job_id, department_id from employees
where (department_id=50)
union
select distinct job_id, department_id from employees
where (department_id=20);

```

100 %

Results Messages

|   | job_id   | department_id |
|---|----------|---------------|
| 1 | AD_ASST  | 10            |
| 2 | MK_MAN   | 20            |
| 3 | MK_REP   | 20            |
| 4 | SH_CLERK | 50            |
| 5 | ST_CLERK | 50            |
| 6 | ST_MAN   | 50            |

Query executed successfully. LENOVO-PC (13.0 RTM) Lenovo-PC\win (54) Human\_Resources 00:00:00 6 rows

4. Crear un reporte que muestre que liste los códigos de los empleados y los puestos de todos aquellos empleados que tienen el mismo puesto que en el momento en el que fueron contratados por la empresa, cambiaron de puestos y luego volvieron al puesto anterior. Utilizar el operador INTERSECT.

```

select employee_id, job_id from employees
intersect
select distinct employee_id, job_id from job_history;

```

Scripts - Consulta...Lenovo-PC\win (54) X Script - Base de D...Lenovo-PC\win (56)

```

/*
EJERCICIO 4

Crear un reporte que muestre que liste los códigos de los empleados y los puestos de todos aquellos
empleados que tienen el mismo puesto que en el momento en el que fueron contratados por la empresa,
cambiaron de puestos y luego volvieron al puesto anterior. Utilizar el operador INTERSECT.*/

select employee_id, job_id from employees
intersect
select distinct employee_id, job_id from job_history;

```

100 %

Results Messages

|   | employee_id | job_id  |
|---|-------------|---------|
| 1 | 176         | SA_REP  |
| 2 | 200         | AD_ASST |

Query executed successfully. LENOVO-PC (13.0 RTM) Lenovo-PC\win (54) Human\_Resources 00:00:00 2 rows

5. El departamento de Recursos Humanos requiere un reporte que muestre lo siguiente:

- Apellidos y códigos de departamentos de todos los registros de la tabla empleados sin importar si pertenecen a uno o ningún departamento.
- Código de departamentos y nombres de departamentos de la tabla DEPARTAMENTOS inclusive si no existiese ningún empleado en ese departamento

Ambos requerimientos se deben mostrar en un mismo resultado. Utilizar el operador UNION ALL.

Para lo cual se observa el siguiente código:

```
select last_name, department_id, null from employees union select null, department_id, department_name from departments;
```

The screenshot shows a SQL script editor with a script titled "EJERCICIO 5". The script contains a comment in Spanish explaining the requirements for a report from the HR department. The SQL query is: `select last_name, department_id, null from employees union select null, department_id, department_name from departments;`

Below the script, the "Results" pane displays the output of the query. It shows a table with three columns: `last_name`, `department_id`, and `(No column name)`. The first column contains 14 rows of `NULL` values, corresponding to the 14 departments listed in the second and third columns. The departments are: Administration, Marketing, Purchasing, Human Resources, Shipping, IT, Public Relations, Sales, Executive, Finance, Accounting, Treasury, Corporate Tax, and Control And Credit.

The status bar at the bottom indicates: "Query executed successfully. | LENOVO-PC (13.0 RTM) | Lenovo-PC\win (54) | Human\_Resources | 00:00:00 | 132 rows".

|    | last_name | department_id | (No column name)   |
|----|-----------|---------------|--------------------|
| 1  | NULL      | 10            | Administration     |
| 2  | NULL      | 20            | Marketing          |
| 3  | NULL      | 30            | Purchasing         |
| 4  | NULL      | 40            | Human Resources    |
| 5  | NULL      | 50            | Shipping           |
| 6  | NULL      | 60            | IT                 |
| 7  | NULL      | 70            | Public Relations   |
| 8  | NULL      | 80            | Sales              |
| 9  | NULL      | 90            | Executive          |
| 10 | NULL      | 100           | Finance            |
| 11 | NULL      | 110           | Accounting         |
| 12 | NULL      | 120           | Treasury           |
| 13 | NULL      | 130           | Corporate Tax      |
| 14 | NULL      | 140           | Control And Credit |